(12)

ADVANCED DECISION SYSTEMS

Final Report
ADS TR-1048-03

January 1986

DISTRIBUTED HYPOTHESIS TESTING IN DISTRIBUTED SENSOR NETWORKS

FINAL REPORT

C.Y. Chong
S. Mori
K.C. Chang
D.S. Spain
R.P. Wishner

Sponsored by

Defense Advanced Research Projects Agency (DoD)
DARPA Order No. 4272/4

Under Contract No. MDA903-83-C-0372 issued by Department of Army, Defense Supply Service - Washington, Washington, D.C., 20310

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

201 San Antonio Circle, Suite 286
Mountain View, California 94040-1289
415/941-3912 FAX: 415/949-4029

Formerly: Advanced Information & Decision Systems

Final Report
ADS TR-1048-03

January 1986

DISTRIBUTED HYPOTHESIS TESTING IN DISTRIBUTED SENSOR NETWORKS

FINAL REPORT

Prepared for:

DARPA
Attn: TIO/Admin.
1400 Wilson Boulevard
Arlington, VA 22209

and

Defense Supply Service - Washington
Washington, DC 20310

Prepared by:

C.Y. Chong
S. Mori
K.C. Chang
D.S. Spain
R.P. Wishner

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS None
2a. SECURITY CLASSIFICATION AUTHORITY N/A	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) ADS-TR-1048-03	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Advanced Decision Systems 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Defense Supply Service - Washington
6c. ADDRESS (City, State and ZIP Code) 201 San Antonio Circle, Suite 286 Mountain View, CA 94040	7b. ADDRESS (City, State and ZIP Code) Room 1D 245, The Pentagon Washington, DC 20310
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Defense Advanced Research Projects Agency 8b. OFFICE SYMBOL (If applicable) IPTO	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER MDA903-03-C-0372
8c. ADDRESS (City, State and ZIP Code) 1400 Wilson Boulevard Arlington, VA 22209	10. SOURCE OF FUNDING NOS.
11. TITLE (Include Security Classification) Distributed Hypothesis Testing in Distributed Sensor Networks	

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
62708E	Program Code: 3T10	ARPA Order No.: 4272/4	

12. PERSONAL AUTHOR(S)
Chong, C.Y., Chang, K.C., Mori, S., Spain, D.S., Wishner, R.P.

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Yr., Mo., Day)	15. PAGE COUNT
Final	FROM 7/15/83 TO 1/31/86	1986 January	134

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB. GR.
15	03	
17	02	

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Distributed Sensor Networks, Distributed Estimation, Distributed Multitarget Tracking, Information Fusion, Group Tracking, Distributed System Testbed Design, Acoustic Tracking, Attribute-Based Tracking

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This report presents research results on distributed situation assessment in distributed sensor networks (DSNs). Information fusion algorithms for hypothesis formation and evaluation are presented. The algorithms are based on the concept of an information graph and issues that only consistent hypotheses are formed and information is not used redundantly. DSNs where the nodes have sensors observing different attributes are also considered. A theory for tracking groups of targets has been developed based on a Bayesian Theory for multitarget tracking. The general approach is used to develop algorithms for tracking with various types of sensors including acoustic. Some simulation results are presented.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED XX SAME AS RPT. ☐ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE NUMBER (Include Area Code) / 22c. OFFICE SYMBOL

DD FORM 1473, 83 APR — EDITION OF 1 JAN 73 IS OBSOLETE

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION AND SUMMARY

This technical report describes research performed on the distributed processing of sensor data for situation assessment in a distributed sensor network (DSN). This research was performed at Advanced Decision Systems under the contract entitled "Distributed Hypothesis Testing in Distributed Sensor Networks".

1.1 DSN PROBLEM DESCRIPTION

We assume a system structure as in Figure 1-1. There is a system of distributed sensor/processor nodes. Each node may have one or more sensor types, and the sensors from different nodes may have overlapping coverage. The sensors collect data from the environment and pass them on to the processors (processing nodes). The processing nodes process the sensor data and communicate with other nodes through the communication network to obtain an assessment of the state of the world. It is generally assumed that no single node possesses complete information and each node may have a different world model. The processing nodes may also control the sensors to improve on the performance of the overall system.

A distributed sensor network can be used for many applications. We are particularly interested in a DSN which is used for the tracking and classification of multiple targets. The target environment is assumed to be dense, so that determining the origins of the measurements in a particular sensor report is not always easy. The problem is further complicated by the presence of false alarms and missing target reports. In such a network, tracking and classification is highly dependent on identifying the right data association hypotheses. Since the nodes in general have access to different information, communication among the nodes can greatly improve the performance of the system. The problem is thus one of distributed hypothesis formation and evaluation, which we can abbreviate as distributed hypothesis testing.

In our previous DSN project we initiated research on the distributed tracking of multiple targets by the nodes of a distributed sensor network. In the following we shall review a model of the processing node that has been studied.

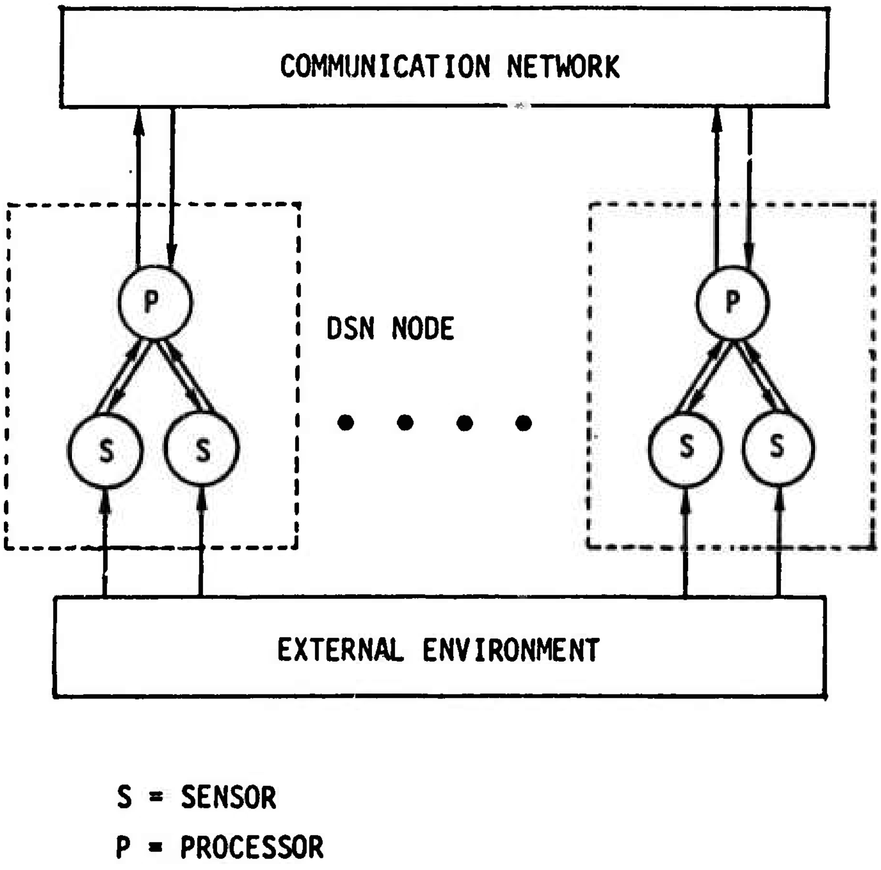


Figure 1-1: Distributed Sensor Network

1.2 PROCESSING NOD IODEL

The processing nodes are the main information processing units in the DSN. Each processing node collects measurements from a set of sensors. Its functions are to process the local sensor data to form an assessment of the state of the world, to combine the information obtained from other nodes with the local information to update its assessment, to distribute information to other nodes, and to performs these functions effectively. These functions are performed in four separate modules within each processing node (see Figure 1-2). In the following we shall discuss the modules in more detail.

1.2.1 Generalized Tracker/Classifier

This module is responsible for the local data processing before any communication with the other nodes takes place. Since the objective of the system under consideration is the tracking and classification of multiple targets, this module is a multitarget tracker. In the previous project, we developed a general theory for multitarget tracking which is implemented in the form of the *Generalized Tracker/Classifier* (GTC). The GTC has the structure shown in Figure 1-3 and itself consists of four modules. The *hypothesis formation* module forms multiple hypotheses from the sensor data, each consisting of a collection of tracks to explain the origins of the measurements in each data set. These hypotheses are then evaluated by the *hypothesis evaluation* module with respect to their probabilities of being true. The *filtering and parameter estimation* module generates state estimates and classifications for each track. It is essential for hypothesis evaluation and can thus be viewed as a submodule. To stay within the computational constraints of each node, the hypotheses are pruned, combined, clustered, etc. This takes place in the *hypothesis management* module. The result of this processing is a set of hypotheses and their probabilities, a collection of tracks corresponding to possible targets and the state distributions of these tracks. These quantities together constitute the information state for multitarget tracking.

1.2.2 Information Fusion

This module combines the local information with information obtained from the other nodes to obtain a new situation assessment. The information from the local nodes consists of the information described above. The information from other nodes is also similar. Information fusion then consists of the following steps

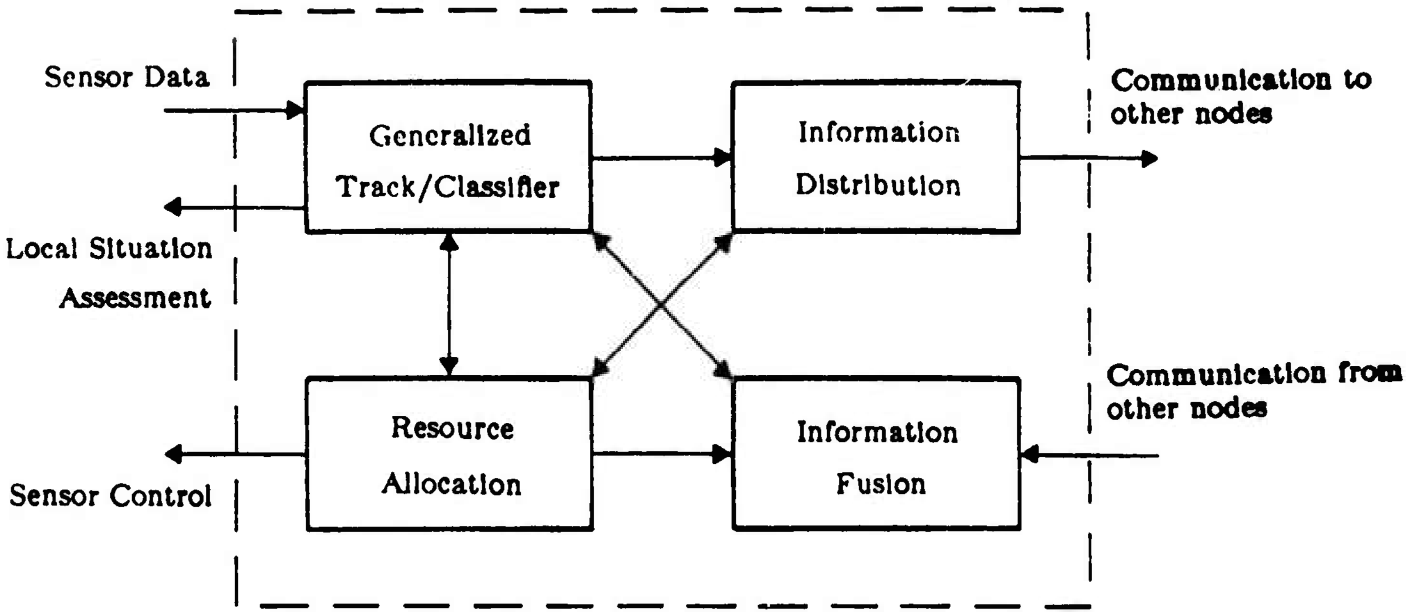


Figure 1-2: Structure of Processing Node

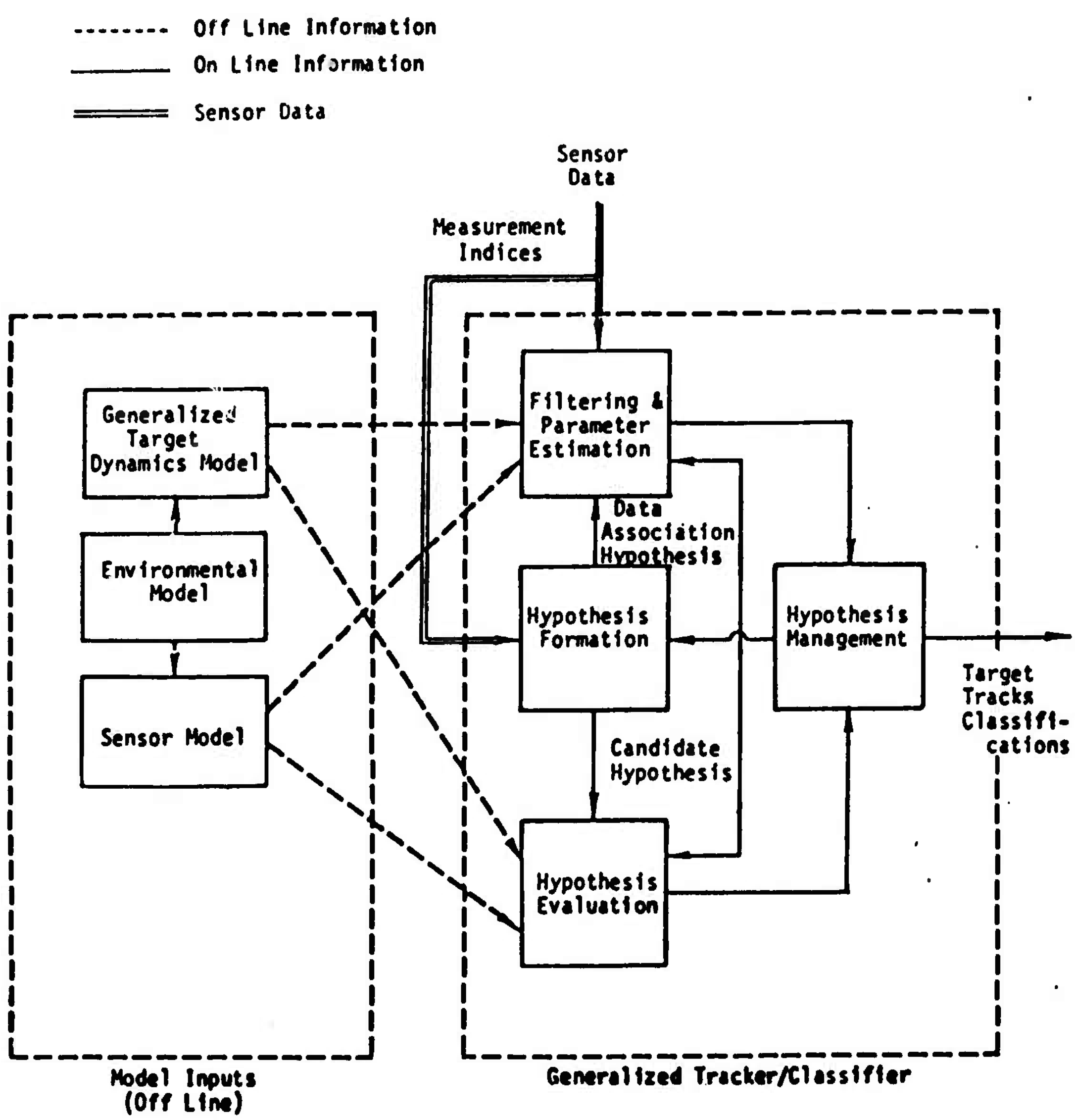


Figure 1-3: Generalized Tracker/Classifier

(see Figure 1-4):

1. *Hypothesis Formation* - Given a set of hypotheses from other nodes, this submodule generates new global hypotheses. Tracks from the hypotheses of different nodes are associated in all possible ways, whether they correspond to the same or different targets.

2. *Hypothesis Evalu* - Each of the hypotheses formed above is then evaluated with respect to its probability of being true. The statistics of the tracks from different hypotheses are used in this evaluation. For example, if two tracks are widely apart in their position or velocity distributions, they are more likely to have come from different targets than the same target.

3. *Hypothesis Management* - This is again needed to make computation feasible within the available resources.

1.2.3 Information Distribution

This module decides what information is to be transmitted, who gets the information, and when it should be communicated. It thus specifies the information available to each node at any time, i.e., the information structure of the system. Information distribution can be fixed a priori for simple systems, or it can be highly adaptive to the information needs in the system.

1.2.4 Resource allocation

This module allocates the resources under the control of the processing node to maintain or improve the performance of the system. Some typical resources include sensor resources and processing resources. Both resource allocation and information distribution can affect the information available in the network. Thus their activities should be coordinated.

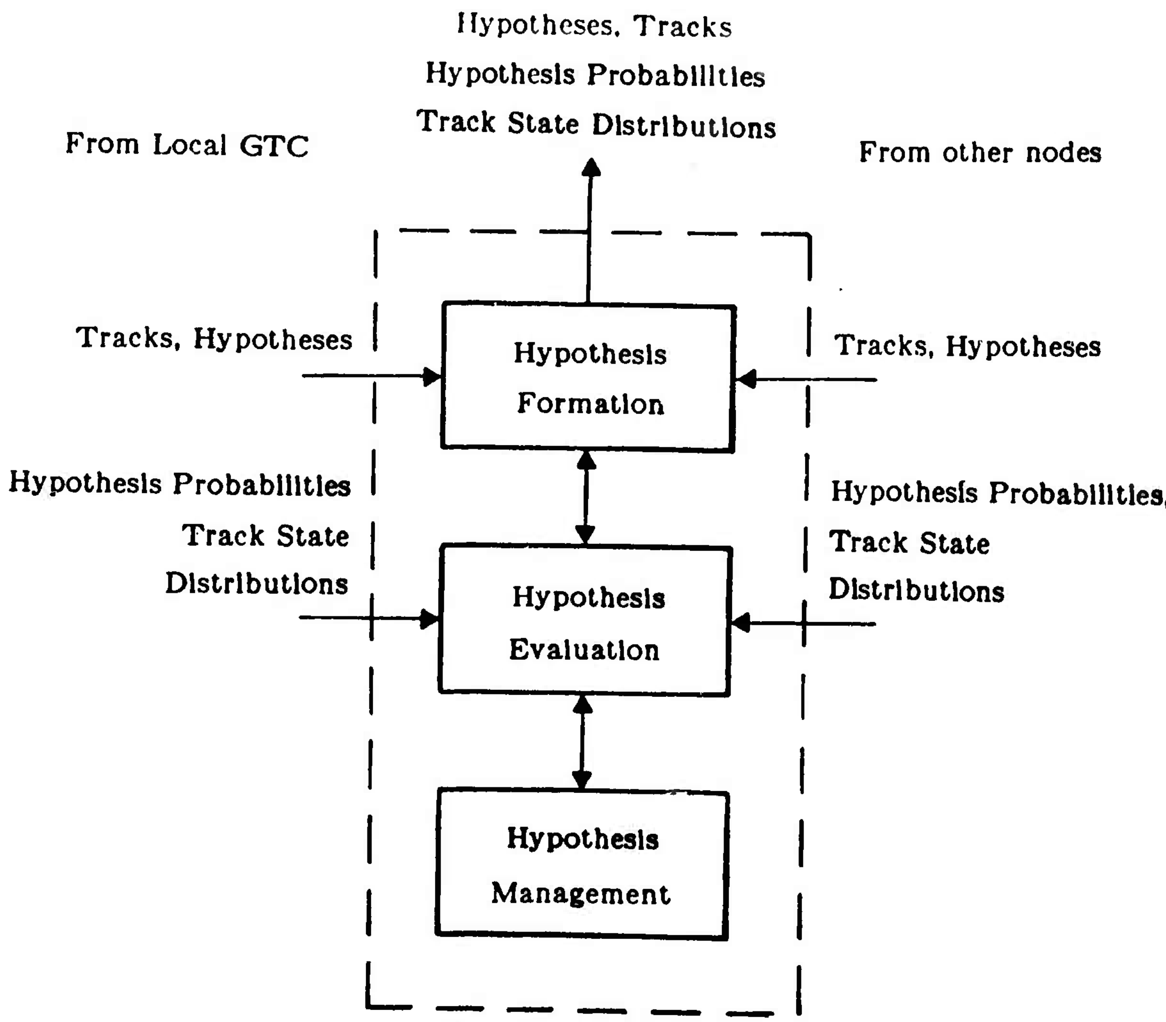


Figure 1-4: Information Fusion

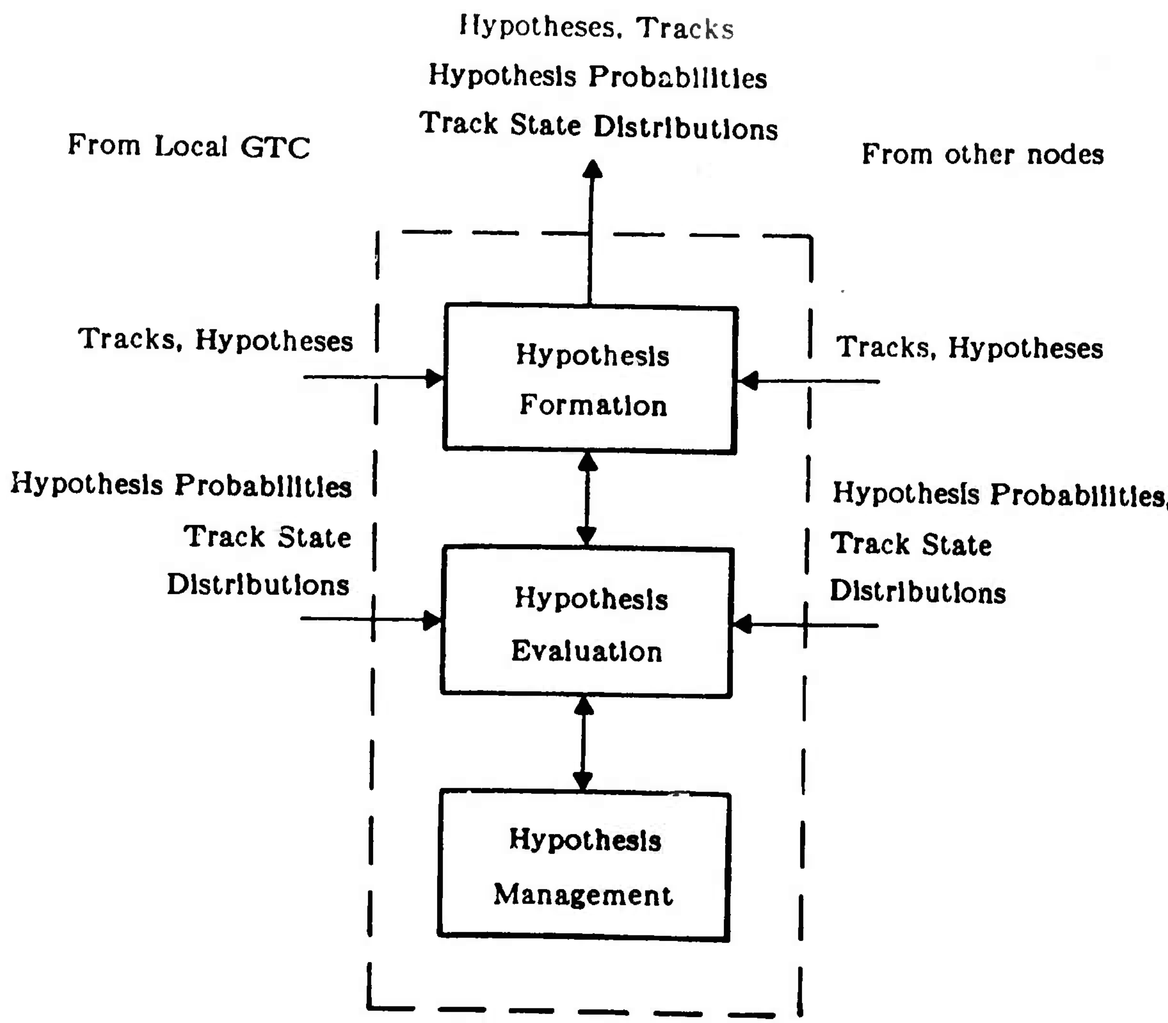


Figure 1-4: Information Fusion

1.3 PROJECT GOALS

Many technical issues have to be addressed before DSNs can be designed, built and operated to achieve their military potential. Such issues include the representation and processing of hypotheses, information fusion, communication strategies, resource allocation, adaptation, system architecture, etc. In our previous DSN project, we successfully addressed some of these issues. The goal of our current effort was to further advance the state of the art in distributed hypothesis testing techniques in DSNs. This would provide more insight as to how a DSN should be designed. Specifically, we intended to accomplish the following technology goals:

1. Develop intelligent distributed algorithms applicable to a wide range of situations such as different network configurations, sensor types, target models; such algorithms should also be adaptive to changing network conditions and make efficient use of sensor resources.

2. Evaluate and adapt these algorithms for real-time implementation.

3. Design experiments to test and evaluate the algorithms in a more realistic scenario such as that used by the Lincoln Laboratory test-bed.

Along with these technology goals, our plan was to develop a simulation environment to test the algorithms experimentally on different scenarios.

1.4 PROJECT ACCOMPLISHMENTS

There were two parts to our research effort. The first consisted of development of algorithms for a DSN and the other was concerned with the development of a simulation environment to test the algorithms and to evaluate the performance of the system experimentally. In the following we discuss both the theoretical and experimental results.

We extended the results of our previous DSN project and developed information infusion algorithms for DSNs with arbitrary communication patterns among the nodes. The key problems are the formation of possible (or meaningful) global hypotheses from a group of local hypotheses and the evaluation of their probabilities. A set of local hypotheses can be inconsistent so that they cannot be fused to form a global hypothesis. The local probabilities of the local hypotheses may depend on common information which needs to be identified. In the previous project, we developed fusion algorithms assuming broadcast communication. In the current project, we obtained fusion algorithms for arbitrary communication. The algorithms are based on modeling the events in the DSN by means of an information graph. To use these algorithms, the histories of the hypotheses and tracks have to be part of the information communicated. Then each node can determine the fusability of the hypotheses and tracks and the common information which has to be removed in evaluating the hypotheses. Information distribution strategies were also considered. These include strategies which depend only on the local information state and those which model the behavior of other nodes.

The theory of multitarget tracking was extended to handle targets with a structured state space and dissimilar sensors which observe different components in the target state. The resulting GTC for processing of local sensor data and the information fusion algorithms are very similar to the usual case. However, a multilevel hypothesis formation and evaluation processing architecture is often possible. Consider a network with two nodes. Each node would form hypotheses based on the local measurements and the tracks would be described in the local feature space. During the fusion process, knowledge on the relationship between the features would be used to generate higher level target tracks from the local feature level tracks. Hypothesis evaluation would then be carried out. As an example, consider the tracking of vehicles. Suppose one sensor node measures only the tread/wheel feature and the location. Feature tracks from this node would consist of wheeled or tread vehicles over time. Suppose another sensor node measures only the location and whether the vehicle has gun or no gun. Tracks generated would consist of gunned or gunless vehicles over time. During fusion, one would use the fact that a vehicle with a gun and tread is a tank, a vehicle with neither gun nor tread is a truck, etc.

In the previous DSN project, we concentrated on independent targets. **In this project, we investigated multitarget tracking on structured sets of targets.** These include targets which move in groups. One example would be planes flying in formation. Another more complex example consists of military force structures. A division would consist of regiments each of which consists of battalions, and so on. The tracking and identification of such structured targets is important but not much systematic treatment is available. The problem is also interesting in a distributed framework since the observations at different nodes may be at different levels and targets are no longer independent. We developed models for structured sets of targets, and the notions of multilevel tracks and hypotheses. They are generalizations of our previous work on multi-target tracking which may be viewed as having a single level of targets. Centralized algorithms for evaluating multilevel hypotheses were obtained. When restricted to two levels with targets moving in independent and identically distributed groups, our results resemble those in single level tracking except the targets in the level are the groups themselves. The main difficulty in implementing these algorithms is in the combinatorics, which becomes more severe with more levels. Thus more practical methods for hypothesis evaluation have to be found. These results can serve as a starting point for finding distributed versions of the algorithms.

As part of the DARPA DSN program, M.I.T. Lincoln Lab. has performed research on the tracking of low flying aircraft using acoustic sensors. **A DSN test bed has been developed and used to test and demonstrate DSN techniques and technology.** The emphasis of the research at Lincoln has been to demonstrate that a DSN is feasible via the construction of a complete (hardware and software) system. Our emphasis, on the other hand, has been the development of general algorithms to detect and track targets in difficult scenarios involving high target density, high false alarm rates, and poor detection conditions. To illustrate the applicability of this general multiple hypothesis to acoustic tracking, we considered to design of experiments using the Lincoln Lab. acoustic tracking scenario. Possible system architectures, and simulation scenarios were investigated with inputs from Lincoln Lab. In addition, we adapted the general distributed tracking algorithm to acoustic tracking. Because of the special features of acoustic sensors (such as azimuth only measurements, acoustic propagation delay), some modifications were made to the general algorithm.

The other part of our research effort was concerned with the development of the simulation environment. Since an analytic evaluation of the algorithms and the system performance is difficult our approach is to perform simulation studies. **We developed a simulation system consisting of four DSN nodes with communication patterns which can be specified arbitrarily.** Our eventual goal for the simulation environment is that it should allow rapid construction of scenarios and rapid development of the DSN system design itself. Also, the environment should be flexible enough to handle various types of processing within each DSN node, including the Bayesian analytic algorithms such as the GTC which have been developed thus far as well as other Artificial Intelligence (AI) based algorithms. Some limited experimentation on this system was performed. The results demonstrate that the nodes can perform better through communication.

1.5 REPORT ORGANIZATION

The rest of this report is organized as follows. In Section 2, we present information fusion algorithms assuming arbitrary communication among the nodes. The algorithms are based on an information graph model of the DSN. Section 3 contains results on tracking using dissimilar sensors. Section 4 presents some algorithms to handle structured sets of targets. In Section 5 the design of experiments for acoustic tracking is discussed. The modification of the general algorithms to handle acoustic sensors is described. Section 6 presents some experimental results with our simulation system.

2. INFORMATION FUSION FOR ARBITRARY COMMUNICATION

In this section we present algorithms used by each node to fuse the information received from the other nodes with the local information to obtain an updated situation assessment. In [1] fusion algorithms for a broadcast communication pattern were presented. The results of this section extend those algorithms to arbitrary communication patterns. In Section 2.1 we describe the information fusion problem in the context of hypothesis formation and evaluation in multitarget tracking. In Section 2.2 a model for information fusion in terms of an information graph is given. Section 2.3 describes the hypothesis formation and evaluation algorithms assuming arbitrary communication.

2.1 THE INFORMATION FUSION PROBLEM

In the following we state the information fusion problem faced by each node in the DSN with emphasis on the relevant issues in multitarget tracking. The formalism is based on the theory of multitarget tracking developed in the previous DSN project [1], [2], and [3].

2.1.1 Local processing

The basic unit of information in the DSN is a *sensor report* $z(t,s)$. This is the output of a sensor s at a time t and is denoted as $((y_j(t,s))_{j=1}^{N_m(t,s)}, N_m(t,s), t, s)$. The index $k=(t,s)$ identifies the sensor report (by time and sensor) uniquely and is called the *sensor report index* or *data index*. $N_m(k)$ is the number of measurements in the report and $(y_j(k))_{j=1}^{N_m(k)}$ is the actual measurement vector. At any given time, let Z be the *data set* consisting of a set of sensor reports and K be the associated *data index set*, i.e, the set of the indices for all the sensor reports contained in Z. The *measurement index set* corresponding to Z is defined as

$$J = \bigcup_{k \in K} \{1,\ldots,N_M(k)\} \times \{k\}. \tag{2.1}$$

Each element $(j,k)=(j,t,s)$ in this set represents the j-th measurement generated at time t by sensor s. The specific value of the measurement is $y_j(t,s)$. According to the system model introduced in Section 1, each node processes the sensor data as they arrive using the *Generalized Tracker/Classifier* (GTC). The output of the GTC when the data is Z consists of the *information state* $\Sigma(Z)$ defined as

$$\Sigma(Z) = (\mathbf{T}(J), (p_t(x \mid \tau, Z))_{\tau \in \mathbf{T}(J)}, \mathbf{H}(J), (P(\Lambda = \lambda \mid Z))_{\lambda \in \mathbf{H}(J)}, \nu(K))$$

where

- $\mathbf{T}(J)$, the set of *possible tracks* defined on J. Each track τ is a subset of J, i.e., $\tau \subseteq J$ and represents the measurement indices coming from a single target. It is usually assumed that a track cannot have two measurement indices in the same sensor report, or the sensor resolution is such that there are no split measurements. Such tracks are then said to be *possible.*

- $p_t(x \mid \tau, Z)$ is the state distribution for a track. Given the track τ, the set of measurements in Z for a hypothesized target is known. From this the distribution of its state x (position, velocity, classification, etc.) at a time t can be found and is a traditional estimation problem. Normally this would be given in terms of a probability distribution; but if the state can be approximated by a Gaussian random vector, the distribution can be expressed in terms of its mean and covariance.

- $\mathbf{H}(J)$ is the set of *possible data-to-data association hypotheses* defined on J. Each *data-to-data association hypothesis* λ is a possible explanation about the origins of all the measurements in Z. Each hypothesis consists of a set of tracks, i.e., $\lambda = \{\tau_1, \tau_2, \ldots\}$. The number of tracks in λ is the number of targets hypothesized to have been detected in the data set Z. Each track τ is the set of measurement indices from a hypothesized target and any measurement index not included in the hypothesis is hypothesized to be a false alarm. We assume that the sensor resolution is such that there are no merged measurements and thus there are no overlapping tracks in the same hypothesis. The set of hypotheses satisfying this property is said to be *possible.* This represents all *mutually exclusive* and *collectively exhaustive* explanations about the origins of the measurements in Z.

- $P(\Lambda = \lambda \mid Z)$ is the probability of that the true data association Λ is a hypothesis λ given all the measurements in Z. Its computation is the key operation in any multiple hypothesis approach to multitarget tracking and recursive algorithms were given in [1], [2], and [3].

- $\nu(K)$ is the expected number of undetected targets up to and including K. It is important for initiating new tracks. If $\nu(K)$ decreases, the likelihood of any measurement coming from a previously undetected target also decreases.

The information state defined above constitutes a state for multitarget tracking since it contains all the relevant information present in the cumulative data set Z. As long as the information state $\Sigma(Z)$ is known, the GTC can continue to process any new sensor report even though the actual data Z is no longer available. When a report is received from a local sensor, the local tracking data are updated. There are three submodules corresponding to the functions of *hypothesis formation, hypothesis evaluation*, and *hypothesis management.*

The *hypothesis formation* submodule forms new hypotheses from the old hypotheses and the data. Consider a report $z(t,s)$ from sensor s at time t. Each measurement $y_j(t,s)$ in the report may come from a previously detected target, from a new target or a false alarm. At the same time, a previously detected target may be missed (undetected) in the current sensor report. Hypothesis formation thus consists of generating these possibilities starting from the old hypotheses. Constraints imposed by the measurement values and possible predicted states of the old tracks should be used to reduce the number of hypotheses formed whenever possible. As a result of this step, sets of possible tracks $\mathbf{T}(J)$ and possible hypotheses $\mathbf{H}(J)$ are formed.

The *hypothesis evaluation* module is responsible for computing the state distribution $p_t(x \mid \tau, Z)$, the probability of each hypothesis $P(\Lambda=\lambda \mid Z)$ and the expected number of undetected targets $\nu(K)$. Recursive algorithms have been developed for computing these. Suppose $k=(t,s)$ represents a new sensor report and the quantities just before the arrival of this sensor report are denoted by $\bar{Z}$, $\bar{K}$ and $\bar{J}$ respectively. Then

$$P(\Lambda=\lambda \mid Z) = C^{-1} P(\bar{\Lambda}=\bar{\lambda} \mid \bar{Z}) L_{FA}(k,\lambda) \prod_{\tau \in \lambda} L(y(k,\tau),\bar{\tau}) \tag{2.2}$$

where C is a normalization constant, $y(k,\tau)$ is the measurement in the sensor report $z(k)$ associated with track τ. The right-hand side of the equation depends on the following likelihoods:

- Likelihood of false alarms $L_{FA}(k,\lambda)$

- Likelihood of a previously detected track $\overline{\tau}$ detected again as measurement y

$$L(y(k,\tau),\overline{\tau}) = \int p_m(y \mid x)\, p_D(x)\, p_t(x \mid \overline{\tau},\overline{Z})\, \mu(dx) \tag{2.3}$$

- Likelihood of a previously detected track $\overline{\tau}$ missed in the current report

$$L(y(k,\tau),\overline{\tau}) = \int (1-p_D(x))\, p_t(x \mid \overline{\tau},\overline{Z})\, \mu(dx) \tag{2.4}$$

- Likelihood of a target newly detected as y

$$L(y(k,\tau),\overline{\tau}) = \overline{\nu} \int p_m(y \mid x)\, p_D(x)\, p_t(x \mid \emptyset,\overline{Z})\, \mu(dx) \tag{2.5}$$

In the above the target state x is a hybrid variable with a continuous part to model geolocation variables and a discrete part to model classification information. For convenience, we define a hybrid measure μ on the state space to be the direct product of a continuous measure and a discrete measure. Then any integral with respect to this hybrid measure is a sum of integrals over the continuous part of the state space. These likelihoods can be computed at the same time as updating the state estimates of the tracks. When the target and sensor models are such that the linear and Gaussian assumptions are satisfied, most of the quantities involved are available from the Kalman filter calculations. As a result of these calculations, probabilities of hypotheses and track state distributions can be obtained.

The *hypothesis management* submodule controls the growth in the number of hypotheses to make the algorithm implementable. This step is crucial for the successful of the multiple-hypothesis approach. Hypothesis management techniques include pruning away low-probability hypotheses, combining similar hypotheses and decomposing the hypothesis set into independent clusters.

2.1.2 Information Fusion Problem

We assume that each node communicates the information state to the other nodes. Suppose a node receives some messages from the other nodes. It has to fuse or integrate this information with the local information to improve on the local estimate. There are many ways of performing fusion. In our work fusion is based on the following philosophy. The ideal case with the highest performance (but also the highest communication cost) is when the nodes communicate the actual sensor data through the network instead of the processed information. In this case a node would be able to generate an optimal information state based on all the data available. Since in a more realistic DSN only the information states are communicated, an appropriate objective for fusion is to reconstruct the optimal information state based on the information states received from the other nodes. To facilitate further discussion, we call the data available to each node before communication takes place as local data and the maximum data set available after communication as global data. Local and global information states, hypotheses, tracks, etc. are all defined analogously.

There are thus two steps to the fusion process. The first step in the fusion process consists of generating the possible track and hypothesis sets based on the global data from the local tracks and hypotheses. Since the local data are the part of the global data available to the nodes at the given times, the global tracks and hypotheses when restricted to the local data should give the local tracks and hypotheses. This implies that a certain combination of local tracks and hypotheses should not be fused, i.e., there may not exist global tracks and hypotheses for given sets of local tracks and hypotheses. In Figure 2-1, the two tracks τ_1 and τ_2 are two local tracks maintained at two different nodes. They cannot be fused since the resulting global track would have two different measurements in the same sensor report 1, thus violating the no split measurement assumption. On the other hand, τ_1 and τ_3 can be fused to yield a global track $\tau_1 \cup \tau_2$. The interpretation of this global track is that the measurements in both tracks τ_1 and τ_3 come from the same target. Tracks τ_1 and τ_4 can also be fused. However, they do not have to be and in that case the two tracks correspond to two different targets. The fusability question also needs to be addressed at the hypothesis level. Each local hypothesis is a possible explanation about the origins of the local measurements. Thus if the local hypotheses are incompatible, they cannot be fused to form a global hypothesis. This is illustrated in Figure 2-2 where each node i has two local hypotheses λ_i^j, $j=1,2$ derived from the two common hypotheses λ^j, $j=1,2$. Since λ^1 and λ^2 are mutually exclusive, the local

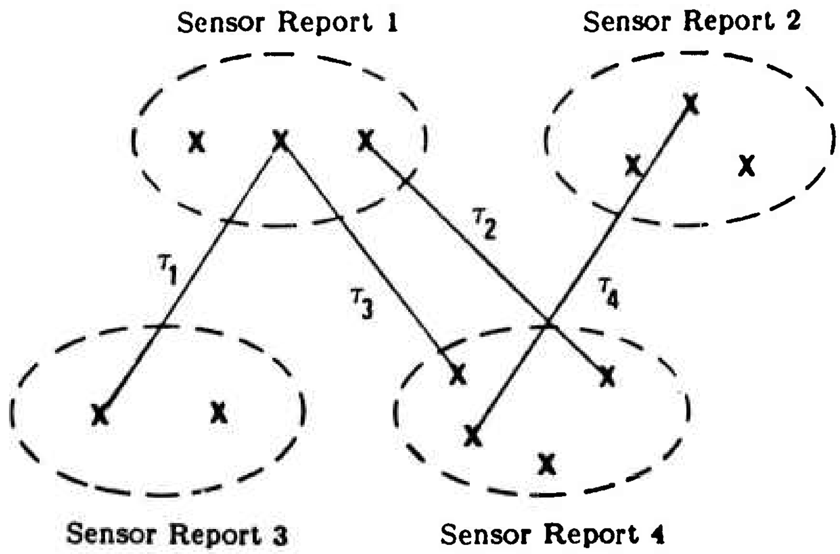


Figure 2-1: Fusability of Tracks

hypotheses λ_1^2 and λ_2^1 are not fusable.

The second step in the fusion process consists in generating the state distributions of the global tracks and the probabilities of the global hypotheses using the local distributions and probabilities. If the nodes communicated in the past, the local statistics would not be independent. A key problem in fusion is to identify the common information shared by the nodes and make sure it is not used more than once in generating the global statistics.

2.2 INFORMATION GRAPH

In performing information fusion, it is necessary to identify the information available to the nodes in the network at various times and how the information of one node at one time is related to that of another node at a different time. For example, whenever two nodes communicate some common information is shared between the nodes. The existence of this shared information would have to be recognized in any future information fusion. Specifically, before any global hypothesis can be generated, the fusability of the local hypotheses have to be

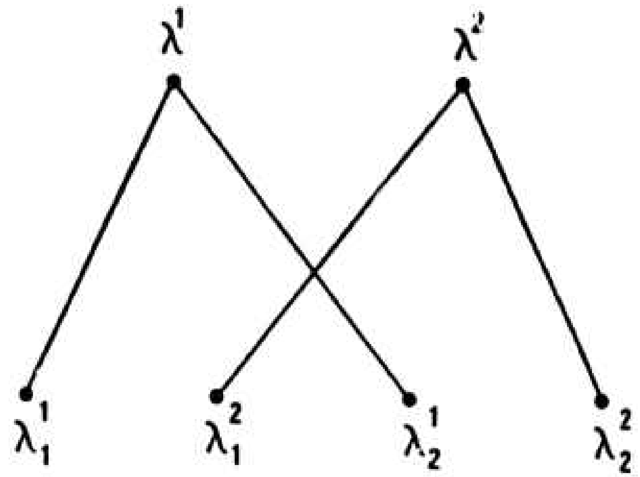


Figure 2-2: Fusability of Hypotheses

checked based on their histories. Furthermore, when the probabilities of the hypotheses are to be evaluated, the common information should only be used once. This necessitates tracking the histories of the communication and can be accomplished conveniently using the information graph. The information graph introduced below can also be viewed as an abstract model for a DSN.

2.2.1 Information graph model

We assume that there is a set of processing nodes called N. Each node n in N receives data from a set of sensors called S_n such that $S_n \cap S_{n'} = \emptyset$ for $n \neq n'$, i.e., each sensor s only reports to one processing node. Let $\mathbf{S} = \bigcup_{n \in \mathrm{N}} S_n$ be the set of all sensors. If a sensor s generates a report at time t with value z, the report is denoted as (z,t,s) or simply $z(t,s)$. Each sensor report is the basic unit of information and the set of all such reports is denoted by $\mathbf{Z}$ called the *total information or data set.* Each sensor report is indexed by $k=(t,s)$, i.e., the time t when it is generated and the sensor s responsible for its generation. The set of all such indices is called the *total data index set* and denoted as

$$\mathbf{K} = \{(t,s) \mid (z,t,s) \in \mathbf{Z} \text{ for some } z\} \tag{2.6}$$

At any one time, a node's information may consist of only a subset Z of $\mathbf{Z}$. Such a Z is called a *partial information set* or *partial data set*, or simply *information set* or *data set*. For each Z there is a K corresponding to the data indices in Z.

The sensors send the data instantaneously to the nodes as soon as they are generated. The communication among the nodes can be characterized by the *communication schedule* **C** which is a subset of $\mathbf{T}\times\mathbf{T}\times\mathbf{N}\times\mathbf{N}$. An element (t,t',n,n') means that the communication transmitted at time t by node n is received at time t' by node n'.

The information at each sensor or node in the DSN is affected by four types of events. The nature of the events, the times at which they occur and the nodes affected are given below:

1. Sensor observation and transmission -- $\mathbf{I}_{ST} = \mathbf{K} \times \{ST\}$

2. Sensor data received at node -- $\mathbf{I}_{SR} = \mathbf{K} \times \{SR\}$

3. Transmission of communication by node --
   $\mathbf{I}_{CT} = \{(n,t,CT) \mid (t,t',n,n') \in \mathbf{C}\}$

4. Reception of communication by node --
   $\mathbf{I}_{CR} = \{(n,t,CR) \mid (t',t,n',n) \in \mathbf{C}\}$

Let **I** be defined as

$$\mathbf{I} = \mathbf{I}_{ST} \cup \mathbf{I}_{SR} \cup \mathbf{I}_{CT} \cup \mathbf{I}_{CR} \tag{2.7}$$

I constitutes all the significant events in the network and forms the set of information nodes (not DSN nodes) in the information graph. To represent the relation between these nodes, we define a partial order (antisymmetric, reflexive and transitive binary relation) $\leq$ on **I** as follows: for any i and i' in **I**, $i \leq i'$ if $i = i'$ or there is a communication path from i to i'. The *information graph* on the system is then the ordered set $(\mathbf{I},\leq)$. By using the graph we can determine how the information in the system flows. In particular, it is easy to find the history of the information at a certain node. As we shall see later, this is useful for the purpose of information fusion.

Figure 2-3 show the information graph for broadcast communication. At a given time all the nodes communicate to each another so that they all have the same information after that. Figure 2-4 shows the information graph for a cyclic communication system. The system consists of three nodes N={1,2,3} collecting data from the three sensors S={1,2,3}, respectively at the times $\ldots, t_{ST}, t_{ST}+t_d, \ldots\ldots$ The nodes transmit to the other nodes periodically according to the pattern shown in Figure 2-4 at times $\ldots, t_{CT}, t_{CT}+t_d, \ldots$ and the messages are received at the times $\ldots, t_{CR}, t_{CR}+t_d, \ldots$ It is assumed that $t_{ST} < t_{CT} < t_{CR}$.

For each information node i in the information graph, the maximum amount of information available is the sensor data that would be received if they had been communicated in the network. Thus associated with each node i the *(maximum) data index set* K_i and the *(maximum) information set* Z_i are defined as follows:

$$K_i = \{k \in \mathbf{K} \mid (k, ST) \leq i\} \tag{2.8}$$

$$Z_i = \{(z, k) \in \mathbf{Z} \mid k \in K_i\}. \tag{2.9}$$

As stated before, our philosophy is to assume that each node tries to reconstruct the best estimate as if all sensor data are transmitted. Thus from now on the information available at each node i is assumed to be Z_i with the data index set K_i.

The following observations are quite obvious from the definitions:

1. $K_i = \{k \in \mathbf{K} \mid (k, SR) \leq i\}$ for all i in **I**.

2. $K_i \subseteq K_{i'}$ if $i \leq i'$. (The information of a node always includes that of any predecessor node.)

3. $K_i = \bigcup_{i' \leq i} K_{i'}$ for all i in **I**. (The information at a node is the union of that of the predecessors.)

4. $K_i = \bigcup_{i' \mapsto i} K_{i'}$ for all i in **I**, where $i' \mapsto i$ means that i' is the immediate predecessor of i. (One needs only to consider the immediate predecessors of i in generating the information available to i.)

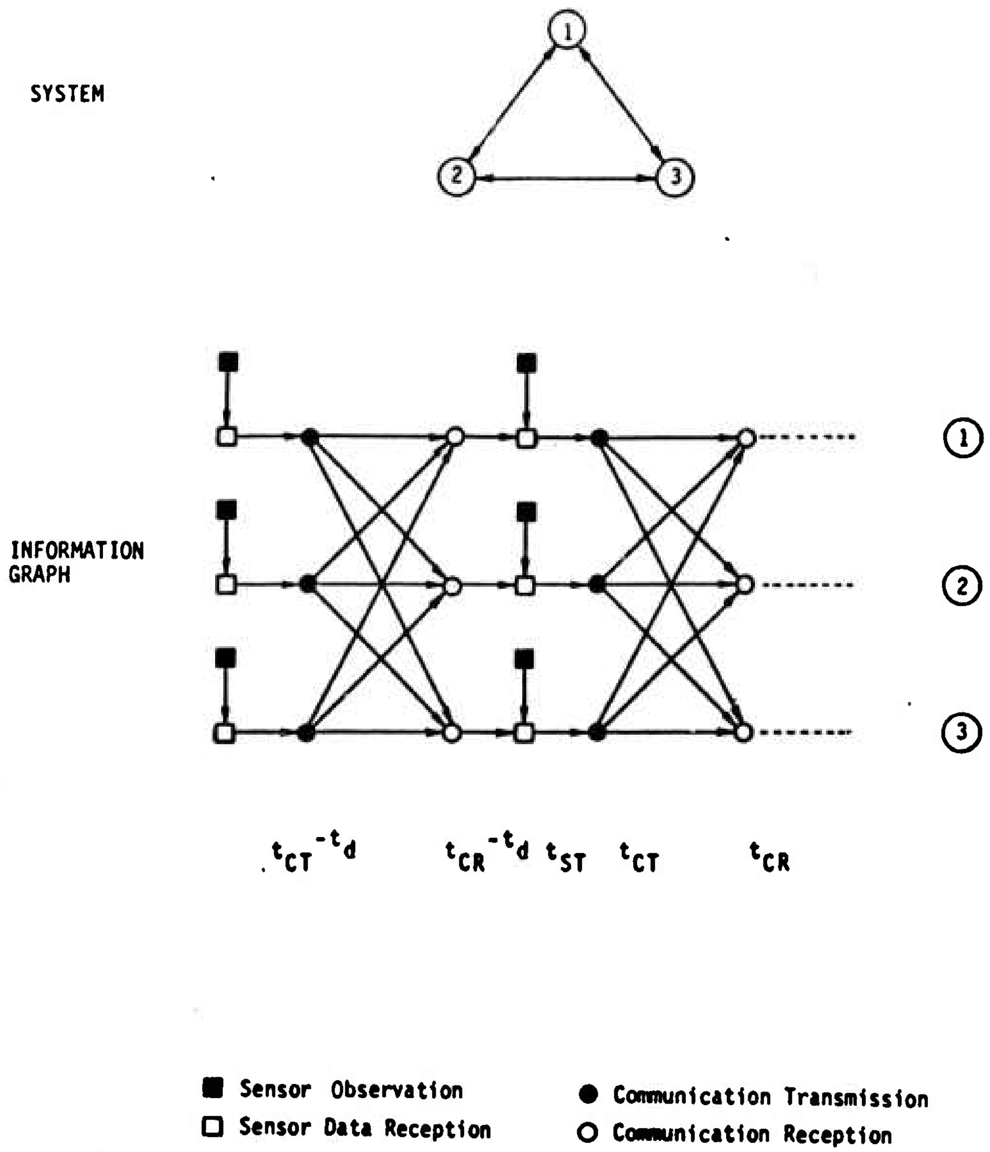


Figure 2-3: Broadcast Communication

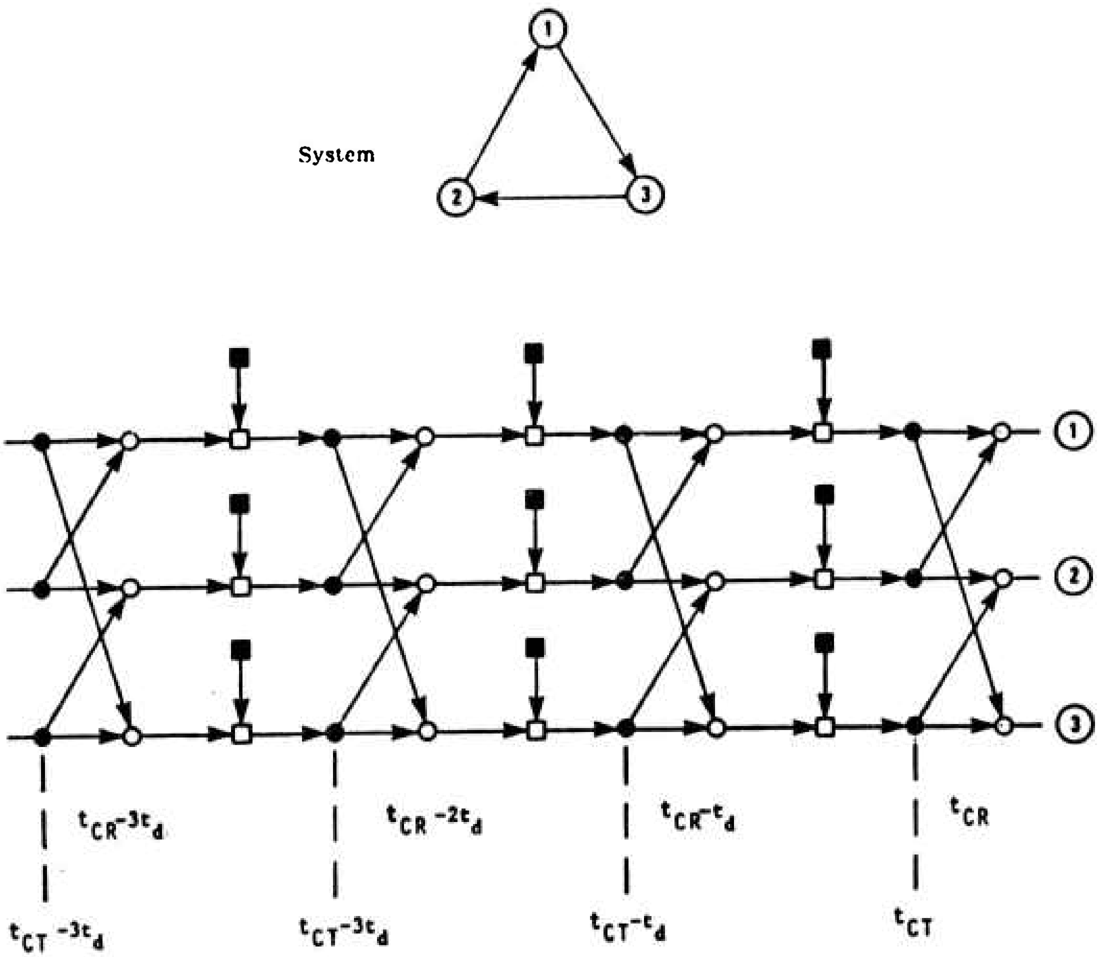


Figure 2-4: Cyclic Communication

Since there is a one-to-one correspondence between K and Z, a similar set of observations can be made for Z.

1. $Z_i = \{Z \in \mathbf{Z} \mid (Z, SR) \leq i\}$ for all i in I.

2. $Z_i \subseteq Z_{i'}$ if $i \leq i'$.

3. $Z_i = \bigcup_{i' \leq i} Z_{i'}$ for all i in I.

4. $Z_i = \bigcup_{i' \mapsto i} Z_{i'}$ for all i in I.

Consider an information node $i_0 \in \mathrm{I}_{CR}$. This represents the event that communication from other nodes is received. Let I be the set of immediate predecessor nodes for i_0. The fusion problem is to find the information state of i_0 using the information states of the nodes in I (and those of other predecessor nodes of I, if necessary). As mentioned before, it is important to identify the common information in the data represented by I. This turns out to be

$$\bigcap_{i \in I} K_i = \bigcup_{i' \in C(I)} K_{i'} \tag{2.10}$$

where

$$C(I) = \{i' \in \mathbf{I} \mid i' \leq i \text{ for } \forall\, i \in I\} \tag{2.11}$$

is the set of common predecessors for all the nodes in I. Equation (2.10) states that the common information shared by the nodes in I is the union of the information of the common predecessor nodes of I. In fact, based on the observation (4) above, $C(I)$ can be replaced by $C_{\max}(I)$ which is the maximum set in $C(I)$ with respect to the set-inclusion partial order whereby $I_1 < I_2$ when $I_1 \subset I_2$ and $i_1 < i_2$ for all $i_1 \in I_1$ and $i_2 \in I_2$. Then the union needs to be taken only over the set $C_{\max}(I)$, i.e., equation (2.10) becomes

$$\bigcap_{i \in I} K_i = \bigcup_{i' \in C_{\max}(I)} K_{i'} \tag{2.12}$$

If necessary, we can regard $C_{\max}(I)$ as I in equation (2.12) and repeat the process to find the common information shared by all the nodes in $C_{\max}(I)$. This would be used in the following section to develop distributed estimation algorithms.

2.2.2 Distributed estimation

We now consider the distributed estimation problem to illustrate the use of the information graph. Any uncertainty in the origins of the measurements is ignored for the time being. The results would be useful in the next subsection when we consider distributed multitarget tracking.

The state to be estimated is a random vector x. The *a priori* probability density (or distribution) is $p(x)$. The observation generated by a sensor s at time t is $z(t,s)$. The following additional assumptions are needed:

- Both the sensor schedule **K** and the communication schedule **C** are independent of the state x.
- Given x and **K**, each element in **Z** is conditionally independent from each other and has an absolutely continuous transitional probability from state x to measurement.

The distributed estimation problem is then to compute $p(x \mid Z_i)$ for each $i \in \mathbf{I}$. From the definition of I, this needs only to be carried out for the sets $\mathbf{I}_{SR}$ and $\mathbf{I}_{CR}$ since the only activities at the other nodes involve transmission. For an information node in $\mathbf{I}_{SR}$, we have a traditional Bayesian update problem where the conditional probability is updated using the sensor report. We are primarily interested in a problem involving information nodes in $\mathbf{I}_{CR}$. Suppose the information node of interest is i_0 and that the immediate predecessors of i_0 form the set I. Then

$$Z_{i_0} = \bigcup_{i \in I} Z_i \tag{2.13}$$

The objective is the computation of $p(x \mid \bigcup_{i \in I} Z_i)$ in terms of the predecessor probabilities $p(x \mid Z_{i'})_{i' \leq i_0}$. Ideally, one would like to use only the probabilities defined on I, but as we shall see, this is not always possible.

In the appendix of [4], we showed that

$$p(x \mid \bigcup_{i=1}^{n} Z_i) = c \prod_{i=1}^{n} \Big(\prod_{N \in \mathbf{N}_i^n} p(x \mid \bigcap_{j \in N} Z_j) \Big)^{(-1)^{i-1}} \tag{2.14}$$

where c is a normalization constant and

$$\mathbf{N}_i^n = \{N \subseteq \{1,..,n\} \mid \#(N)=i\} \tag{2.15}$$

is the set of all subsets of $\{1,..,n\}$ with i elements. In equation (2.15), $\#(N)$ denotes the number of elements in the set N. For $n=2$, this yields the fusion formula for two nodes:

$$p(x \mid Z_1 \cup Z_2) = c \frac{p(x \mid Z_1) p(x \mid Z_2)}{p(x \mid Z_1 \cap Z_2)} \tag{2.16}$$

Equation (2.15) can be interpreted as follows. Since the probabilities $p(x \mid Z_1)$ and $p(x \mid Z_2)$ both utilize the information contained in $Z_1 \cap Z_2$, the division by $p(x \mid Z_1 \cap Z_2)$ is needed to remove the common information so that it is used only once. Equation (2.14) is just a general form where the probabilities from multiple nodes are to be fused. Unfortunately, in both (2.14) and (2.16) there are still terms involving intersections of the Z_i's. If all these intersections are of the form Z_j for some information node j or empty corresponding to the common a priori information, then equation (2.14) or (2.16) serves as a fusion algorithm. In this algorithm, the conditional probability at the fusion node is a product and ratio of the conditional probabilities defined on a set of predecessor nodes. From the definition of the information graph, all these probabilities can be communicated.

If there is an intersection $\bigcap_{j \in N} Z_j$ which is not equal to $Z_{j'}$ for some $j' \in \mathbf{I}$, then by (2.10) the intersection can be expressed as the union of the information of some information nodes again. Equation (2.14) can then be applied to evaluate the probability $p(x \mid \bigcap_{j \in N} Z_j)$. The process can be repeated until all the probabilities are either conditioned on the information at the individual information nodes or the a priori information. For notational convenience, we represent the a priori information by adding an element $i_\emptyset$ to the set $\mathbf{I}$ of all the information nodes and let $\bar{\mathbf{I}} = \mathbf{I} \cup \{i_\emptyset\}$. Then the extended information graph $(\bar{\mathbf{I}}, \leq)$ is constructed by letting $i_\emptyset$ be the immediate predecessor of all the minimum nodes in the original information graph $(\mathbf{I}, \leq)$. Then we have $Z_{i_\emptyset} = K_{i_\emptyset} = \emptyset$. With this definition it can be shown (see Appendix of [4]) that

$$p(x \mid \bigcup_{i \in I} Z_i) = C \prod_{\bar{i} \in \bar{I}} p(x \mid Z_{\bar{i}})^{\alpha(\bar{i})} \tag{2.17}$$

where $\bar{I} \leq I$ is a subset of $\bar{\mathbf{I}}$, $(\alpha(\bar{i}))_{\bar{i} \in \bar{I}}$ is some index tuple such that $\alpha(\bar{i})$ is a nonzero integer for each $\bar{i}$, and C is the normalizing constant. The set $\bar{I}$ contains all the information nodes which are relevant to fusion at node i_0. $\alpha(\bar{i})$ decides whether the information at node $\bar{i}$ should be added ($\alpha(\bar{i})=1$) or removed ($\alpha(\bar{i})=-1$).

To illustrate the use of this algorithm, let us first consider a broadcast communication pattern of Figure 2-3. For notational simplicity, we would suppress the type of the node in naming the node. Consider the information node (t_{CR}, n). We have

$$\bigcap_{n \in N} Z(t_{CT}, n) = Z(t_{CR} - t_d, n). \tag{2.18}$$

Thus, the fusion algorithm for a node n at time t_{CR} is

$$p(x \mid Z(t_{CR}, n)) = C \prod_{i \in N} \frac{p(x \mid Z(t_{CT}, i))}{p(x \mid Z(t_{CR} - t_d, i))} p(x \mid Z(t_{CR} - t_d, n)) \tag{2.19}$$

where C is a normalizing constant. Each term in the product is the new information contained in the sensor report $z(t_{ST}, i)$.

For the cyclic communication system shown in Figure 2-4, consider node 1 at time t_{CR}. The immediate predecessors of the information node $(t_{CR}, 1)$ are $(t_{CT}, 1)$ and $(t_{CT}, 2)$. Equation (2.16) can thus be used to find $p(x \mid Z(t_{CR}, 1))$. From the information graph of Figure 2-4, the common predecessors of $(t_{CT}, 1)$ and $(t_{CT}, 2)$ consist of the two nodes $(t_{CT} - 2t_d, 1)$ and $(t_{CT} - t_d, 2)$. Thus

$$Z(t_{CT}, 1) \cap Z(t_{CT}, 2) = Z(t_{CT} - 2t_d, 1) \cup Z(t_{CT} - t_d, 2), \tag{2.20}$$

and equation (2.16) can be used to find the probability of the right hand side again. From the information graph,

$$\begin{aligned} Z(t_{CT} - 2t_d, 1) \cap Z(t_{CT} - t_d, 2) &= Z(t_{CT} - 3t_d, 1) \cup Z(t_{CT} - 3t_d, 2) \\ &= Z(t_{CR} - 3t_d, 1). \end{aligned} \tag{2.21}$$

Thus, the algorithm gives for general $i = 1,2,3$

$$p(x \mid Z(t_{CR}, i)) = C \frac{p(x \mid Z(t_{CT}, i))}{p(x \mid Z(t_{CT} - 2t_d, i))} \frac{p(x \mid Z(t_{CT}, [i+1]))}{p(x \mid Z(t_{CT} - t_d, [i+1]))}$$

$$\times p(x \mid Z(t_{CR} - 3t_d, i)) \tag{2.22}$$

where $[i]$ is i modulo 3.

This is in the form of equation (2.17) with five nodes in the set $\overline{I}$. Thus, in addition to its current conditional probability $p(x \mid Z(t_{CT}, 1))$, and $p(x \mid Z(t_{CT}, 2))$ which comes from node 2, node 1 has to store three other probabilities. Note that $p(x \mid Z(t_{CT} - t_d, 2))$ is available to node 1 from earlier communications. This indicates that in a distributed sensor network, knowing the most recent estimate may not be sufficient if one wants to recover the globally optimal estimate.

Our discussion above assumes the fusion algorithm for each node is provided by a system designer based on the information graph. Alternatively, we may assume that the information graph is known to all the DSN processing nodes who then compute the algorithms in a distributed manner. Still another possibility is for each message to contain a history of the nodes and times that it has passed through. Then a fusion node can use the histories of the messages received to construct a partial information graph so that fusion can be performed. This philosophy would be useful for fusion when the communication pattern is not fixed a priori, such as when nodes can vary their communication strategies or have to adapt to system failures.

2.3 FUSION IN MULTITARGET TRACKING

In this section we consider the fusion algorithm for multitarget tracking assuming arbitrary communication pattern. The algorithm is based on the theory of multitarget tracking developed under the previous project [1] and the concept of the information graph. In the previous project [1], the information fusion in multitarget tracking was investigated primarily for broadcast type communication pattern. In this section, we treat the same subject assuming an arbitrary communication pattern which is defined in terms of an information graph.

2.3.1 Problem formulation

In Section 2.1 we introduced the fusion problem in general terms. We now state it more formally in terms of an information graph. Given the communication pattern of the network, an information graph is defined. For each information node i in the graph, there is a data index set K_i and an information set or data set Z_i as defined before. Since we are now interested in multitarget tracking, we have to deal with measurement index sets on which tracks and hypotheses are defined. A measurement index set J_i at an information node i is defined as

$$J_i = \{(j,k) \in \mathbf{J} \mid k \in K_i\}.$$

The activities in a DSN can be represented by the expansion of the nodes in the information graph. Two types of nodes, namely those in $\mathbf{I}_{ST}$ and $\mathbf{I}_{CT}$, involve only communication. For the other two types, namely the ones in $\mathbf{I}_{SR}$ and $\mathbf{I}_{CR}$, information processing is involved. At a node in $\mathbf{I}_{SR}$, the data received from the local sensors are processed by each node using the GTC, producing an information state for the node. For a node $i_0 \in \mathbf{I}_{CR}$, messages are received from other nodes in the DSN and fusion takes place. Let I be the set of immediate predecessor nodes of i_0. For any node i in I, assume the possible tracks $\mathbf{T}(J_i)$ and the possible hypotheses $\mathbf{H}(J_i)$ are known. In addition to these, the local probabilities of the tracks and hypotheses are also given. From the information graph, the measurement index set for the information node i_0 is $J = \bigcup_{i \in I} J_i$. The two specific subproblems in information fusion are then the following:

- (Hypothesis formation) How should node i_0 construct the *possible* (global) track set $\mathbf{T}(J)$ and the *possible* (global) hypothesis set $\mathbf{H}(J)$?

- (Hypothesis evaluation) Suppose the global sets of tracks and hypotheses are formed. How can we evaluate the probability of each hypothesis using the probabilities of the predecessor nodes? Also, how should the state distributions of the tracks be computed?

The two problems would now be discussed separately.

2.3.2 Hypothesis formation

As we discussed before in Section 2.1, not all local tracks and hypotheses can be fused to form meaningful global tracks and hypotheses. Our philosophy behind information fusion is to reconstruct the information state $\Sigma(Z)$ starting from the information states $\Sigma(Z_i)$. This means that two tracks can only be fused if there exists a global track which is consistent with them. This is also the idea behind the fusion of hypotheses. The following are some definitions needed to formalize this concept.

Consider any two measurement index sets J_1 and J_2 with $J_2 \subseteq J_1$. For each track τ in $\mathbf{T}(J_1)$ the *restriction* of the track τ on J_2 is defined as $\tau \cap J_2$, i.e., the track consisting of only those measurement indices in J_2. We usually say that the track τ is a *successor* of its restriction $\tau \cap J_2$ or conversely, $\tau \cap J_2$ is the *prede-*

cessor track of τ. Similarly, for each hypothesis λ in $\mathbf{H}(J_1)$, the *restriction* of the hypothesis λ on J_2 is defined to be

$$\lambda \mid J_2 = \{\tau \cap J_2 \mid \tau \in \lambda\} BS \{\emptyset\} \tag{2.23}$$

i.e., a hypothesis whose tracks are those of λ restricted to J_2. The concepts of predecessor and successor hypotheses can be defined as in tracks.

Let $(J_i)_{i \in I}$ be an arbitrary tuple of measurement index sets where I is an arbitrary nonempty set. (I does not have to be related to the information graph at all.) Then any tuple $(\tau_i)_{i \in I}$ of tracks in $\prod_{i \in I} \mathbf{T}(J_i)$ is said to be *fusable* if there exists a track τ in $\mathbf{T}(\bigcup_{i \in I} J_i)$ such that

$$\tau \cap J_i = \tau_i \tag{2.24}$$

for all $i \in I$. τ is a track obtained by fusing the tracks in the tuple. Similarly any tuple $(\lambda_i)_{i \in I}$ of hypotheses in $\prod_{i \in I} \mathbf{H}(J_i)$ is said to be *fusable* if there exists a hypothesis λ in $\mathbf{H}(\bigcup_{i \in I} J_i)$ such that

$$\lambda \mid J_i = \lambda_i \tag{2.25}$$

for all $i \in I$. Fusability of tracks thus means that there exists a possible global track such that each of the local tracks represents a restriction of the global track to the local measurement indices. Similarly the fusability of the hypotheses means there exists a global hypothesis such that each local hypothesis is a restriction of the global hypothesis to the local measurement index set, or more specifically, the nonempty restrictions of the tracks in the global hypothesis are the local hypotheses.

If the measurement index sets $(J_i)_{i \in I}$ do not intersect, fusability of tracks and hypotheses is trivially assured. When the measurement index sets do overlap, we have to be concerned about the consistency in the tracks and hypotheses. The following rather intuitive conditions for checking fusability are proved in the appendix.

1. Any track tuple $(\tau_i)_{i \in I}$ in $\prod_{i \in I} \mathbf{T}(J_i)$ is fusable if and only if

$$\tau_{i_1} \cap (J_{i_1} \cap J_{i_2}) = \tau_{i_2} \cap (J_{i_1} \cap J_{i_2}) \tag{2.26}$$

for all $(i_1, i_2) \in I \times I$.

2. Any hypothesis tuple $(\lambda_i)_{i\in I}$ in $\prod_{i\in I}\mathbf{H}(J_i)$ is fusable if and only if

$$\lambda_{i_1} \mid (J_{i_1} \cap J_{i_2}) = \lambda_{i_2} \mid (J_{i_1} \cap J_{i_2}) \tag{2.27}$$

for all $(i_1,i_2)\in I\times I$.

These two conditions state that a tuple of tracks (or hypotheses) is fusable if and only if they share common predecessors (in tracks or hypotheses) in the overlapping measurement index set

$$\bar{J} = \cup\{J_{i_1}\cap J_{i_2} \mid (i_1,i_2)\in I\times I \textit{ such that } i_1\neq i_2\} \tag{2.28}$$

To check the conditions described by (2.27) or (2.28), we need to have tracks and hypotheses defined on the set $\bar{J}$. In general, these are not directly available since there may not be any information node with $\bar{J}$ as its measurement index set. However, by using the decomposition algorithm of equation (2.10), we can express the set $\bar{J}$ as the union of the measurement index sets of some predecessor nodes in the information graph. The two fusability conditions of equations (2.26) and (2.27) can be further reduced to the following.

Let i_0 be a communication receiving node and I be the set of all the immediate predecessors of it. For each $(i_1,i_2)\in I\times I$, let $\bar{I}(i_1,i_2)$ be a set of information nodes $\bar{i}$ such that $\bar{i}<i_1$ and $\bar{i}<i_2$, i.e., their common predecessor nodes. Then, we have

1. a necessary and sufficient condition for any track tuple $(\tau_i)_{i\in I}\in\prod_{i\in I}\mathbf{T}(J_i)$ to be fusable is that, for any $(i_1,i_2)\in I\times I$,

$$\tau_{i_1}\cap J_{(\bar{i})} = \tau_{i_2}\cap J_{(\bar{i})} \tag{2.29}$$

for any $\bar{i}\in\bar{I}(i_1,i_2)$, and

2. a necessary condition for any hypothesis tuple $(\lambda_i)_{i\in I}\in\prod_{i\in I}\mathbf{H}(J_i)$ to be fusable is that, for any $(i_1,i_2)\in I\times I$,

$$\lambda_{i_1} \mid J_{(\bar{i})} = \lambda_{i_2} \mid J_{(\bar{i})} \tag{2.30}$$

for any $\bar{i}\in\bar{I}(i_1,i_2)$.

In general, for any two distinct nodes i_1 and i_2, their common predecessor set $\bar{I}(i_1,i_2)$ may not be unique. However, to use the above conditions to test the fusability, we need only to consider the set of all the maximum elements in the set $\{\bar{i} \in \mathbf{I} \mid \bar{i} < i_1 \text{ and } \bar{i} < i_2\}$, i.e., the maximum common predecessor set. Thus in the cyclic communication example of Figure 2-4, a track from the node $(t_{CR},1)$ and one from the node $(t_{CR},2)$ are fusable if and only if they have the same predecessor (or restriction) tracks in both the nodes $(t_{CT}-2,1)$ and $(t_{CT}-1,2)$.

The test defined by (2.29) provides a necessary and sufficient condition for track fusability but equation (2.30) only provides a necessary condition for hypothesis fusability. This is due to the fact that a fusable tuple of tracks produces only one fused track but a fusable tuple of hypotheses may produce more than one hypotheses. The counterexample in Figure 2-5 shows that (2.30) is not a sufficient condition for the hypothesis fusability. In this example, the two hypotheses (λ_1,λ_2) are to be fused. The common predecessors of the nodes 1 and 2 are nodes 3 and 4. It is obvious that $\lambda_1 \mid J_3 = \lambda_2 \mid J_3$ and also $\lambda_1 \mid J_4 = \lambda_2 \mid J_4$,

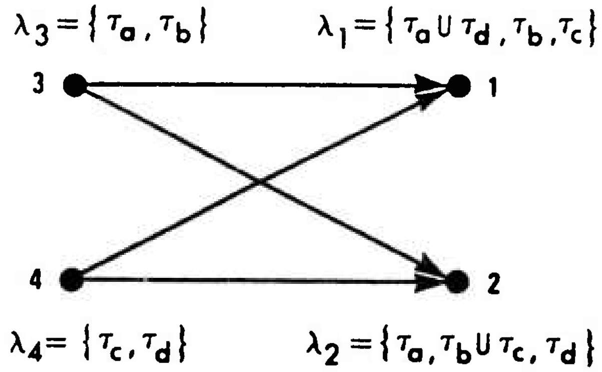


Figure 2-5: Counter Example

thus satisfying the necessary condition of (2.30) for hypothesis fusability. In fact, this is true since both λ_1 and λ_2 are the results of fusing λ_3 and λ_4. However, since

$$\lambda_1 \mid J_3 \cap J_4 \neq \lambda_2 \mid J_3 \cap J_4 \tag{2.31}$$

the hypothesis fusability condition of (2.27) is violated. This is again obvious since λ_1 and λ_2 are mutually exclusive. λ_1 hypothesizes that τ_a and τ_d are from the same target whereas λ_2 hypothesizes that τ_a and τ_d are from different targets.

Although it is not sufficient to determine hypothesis fusability by considering only the predecessors of the hypotheses in the predecessor nodes, the condition (2.30) can be used to eliminate hypotheses for further consideration if they do not have the same predecessor hypothesis in a common predecessor node. Furthermore, the following equivalence condition (proved in the appendix of [4]) relates hypothesis fusability to track fusability.

Hypothesis Fusability Condition. Let $(J_i)_{i \in I}$ be any tuple of measurement index sets and $J = \bigcup_{i \in I} J_i$. Then, any $(\lambda_i)_{i \in I} \in \prod_{i \in I} \mathbf{H}(J_i)$ is fusable with fused hypothesis $\lambda \in \mathbf{H}(J)$ if and only if

1. for any τ in λ, there exists a fusable track tuple $(\tau_i)_{i \in I} \in \prod_{i \in I} (\lambda_i \cup \{\emptyset\})$ such that $\tau = \bigcup_{i \in I} \tau_i$, and

2. for all $i \in I$ and for all $\tau_i \in \lambda_i$, there exists a unique τ in λ such that $\tau_i \subseteq \tau$.

Condition 1 states that every track τ in the hypothesis λ is formed by taking the union of the fusable tracks in the local hypotheses. Condition 2 states that every τ_i belongs to a unique global track in any given global hypothesis.

Hypothesis formation thus consists of the following steps:

1. Use the necessary condition of (2.30) to reduce the candidates for fusable hypothesis tuples

2. Use the track fusability condition of (2.29) to further determine hypothesis fusability

3. Exhaust all possible fusable hypothesis tuples, and for each fusable hypothesis tuple, generate all possible fused hypotheses.

The last step is concerned with the actual hypothesis formation and consists of a two-level procedure. The first level performs hypothesis-to-hypothesis association. The second level carries out the actual track-to-track association to form global tracks from the fusable track tuples.

2.3.3 Hypothesis evaluation

Given the global hypotheses and global tracks constructed from the local hypotheses and local tracks, the objective of hypothesis evaluation is to compute their probabilities and state distributions using the communicated local information. In terms of the information graph, the problem is as follows. Let $i_0=(t, n, CR)$ be a communication receiving node in $\mathbf{I}_{CR}$ and I be the set of all the immediate predecessors of i_0. Let $Z=\bigcup_{i\in I} Z_i$ with K and J be the associated index set and measurement index set. We need to compute the probabilities of all hypotheses, $(P(\Lambda=\lambda \mid Z))_{\lambda\in\mathbf{H}(J)}$, the state distributions of the tracks, $(p_t(x \mid \tau, Z))_{\tau\in\mathbf{T}(J)}$, and the expected number $\nu(K)$ of undetected targets.

We make the standard assumptions on the target and sensor models (see [1] or [2]). In particular, the target models are assumed to the independent and identically distributed Markov processes and the number of targets is Poisson distributed. The sensor measurements generated by sensors at different times are conditionally independent given the target state. In addition to these, we also make the special assumption that the target state is either static or bidirectionally deterministic (which makes it equivalent to a static process). This assumption is needed to make the algorithm more implementable. Later in this section, we should briefly discuss how this assumption can be relaxed. The target state is in a hybrid variable with a continuous part to model geolocation type variables and a discrete part to model classification type information. For convenience, we define a hybrid measure μ on the state space to be the direct product of a continuous measure and a discrete measure. Then any integral with respect to this hybrid measure is a sum of integrals over the continuous part of the state space.

With these assumptions, the following hypothesis evaluation results are derived in the appendix. Let $(\bar{I},\alpha)$ be the pair which satisfies the condition (2.17) of Section 2.2.2. Suppose for each $\bar{i}\in\bar{I}$, the probability $p(\lambda \mid Z_{\bar{i}})$ for each hypothesis λ in $\mathbf{H}(J_{\bar{i}})$, the track state distribution $p(x \mid \tau, Z_{\bar{i}})$ for each track τ in $\mathbf{T}(J_{\bar{i}})$, and $\nu(K_{\bar{i}})$, the expected number of undetected targets are all known. Then for every hypothesis $\lambda\in\mathbf{H}(J)$, the probability of the hypothesis being true is given by

$$P((\Lambda \mid J)=\lambda \mid Z) = C^{-1} \prod_{\bar{i}\in\bar{I}} P((\lambda \mid J_{\bar{i}}) \mid Z_{\bar{i}})^{\alpha(\bar{i})} \prod_{\tau\in(\lambda \mid J)} \tilde{L}(\tau,(Z_{\bar{i}})_{\bar{i}\in\bar{I}}) \tag{2.32}$$

where C is a normalization constant, and

$$\tilde{L}(\tau,(Z_{\bar{i}})_{\bar{i}\in\bar{I}}) = \int \prod_{\bar{i}\in\bar{I}} \tilde{p}(x \mid Z_{\bar{i}},(\tau\cap J_{\bar{i}}))^{\alpha(\bar{i})}\mu(dx) \tag{2.33}$$

is the likelihood of the global track τ. The expected number of undetected targets is given by

$$\nu(K) = \tilde{L}(\emptyset,(Z_{\bar{i}})_{\bar{i}\in\bar{I}}) = \int \prod_{\bar{i}\in\bar{I}} \tilde{p}(x \mid \emptyset, Z_{\bar{i}})^{\alpha(\bar{i})}\mu(dx) \tag{2.34}$$

where

$$\tilde{p}(x \mid \tau, Z_{\bar{i}}) = p(x \mid \tau, Z_{\bar{i}})\nu(K_{\bar{i}})^{\epsilon_{\bar{i}}(\tau)}, \tag{2.35}$$

$$\epsilon_{\bar{i}}(\tau) = \begin{cases} 1 & \text{if } \tau\cap J_{\bar{i}}=\emptyset \\ 0 & \textit{otherwise} \end{cases} . \tag{2.36}$$

The state distribution of the track τ can be updated by

$$p(x \mid \tau, Z) = c^{-1} \prod_{\bar{i}\in\bar{I}} p(x \mid (\tau\cap J_{\bar{i}}), Z_{\bar{i}})^{\alpha(\bar{i})} \tag{2.37}$$

where c is a normalization constant.

We note first of all that hypothesis evaluation depends only on the statistics at the information nodes in the set $\bar{I}$. This is the same set used in distributed estimation and represents the nodes which are relevant for fusion. The function α determines whether the information at a node should be added or subtracted. The hypothesis evaluation formula of (2.32) has a two-level structure. At the higher level, the product of the local hypothesis probabilities evaluates the probability of associating the given set of local hypotheses. The next level consists of the likelihoods of the individual tracks.

Each $\tilde{L}(\tau,(Z_{\bar{i}})_{\bar{i}\in\bar{I}})$ is a track-to-track association likelihood, i.e., the likelihood of associating all the tracks in the local track tuple $(\tau\cap J_{\bar{i}})_{\bar{i}\in\bar{I}}$ with one target represented by the global track τ which is their union. Its evaluation depends on the state distributions of the local tracks. If the tracks have similar state descriptions then the integrand in equation (2.33) will be large, thus resulting in a high likelihood. On the other hand, if the local tracks have state descriptions which are very different, the integrand in (2.33) will be small, resulting in a low likelihood. In equation (2.33), the function $\tilde{p}(x \mid \tau,Z_{\bar{i}})$ is identical to $p(x \mid \tau,Z_{\bar{i}})$, the state distribution for track τ, when the track τ has a nonempty restriction at the node $\bar{i}$. When this is not the case, i.e., the track τ has not been detected yet at $\bar{i}$, the function $\tilde{p}$ is scaled by the expected number of undetected targets and is no longer a probability distribution. It represents some kind of density for undetected targets.

Equation (2.34) computes the expected number of undetected targets by fusing the local track state distributions of the undetected targets. Equation (2.37) is the fusion formula for the global track state distribution. Note that it has the same form as (2.17). This is not at all surprising since given a particular track, computing the state distribution of the target is the usual estimation problem. Thus the fusion algorithm for distribution estimation is an integral part of fusion for multitarget tracking.

2.4 CONCLUSION

In this section, we have described the results of our research on information fusion for multitarget tracking. We have identified two main problems in information fusion assuming arbitrary communication. The first is how to generate meaningful tracks and hypotheses starting from a set of local tracks and hypotheses. The second is how to compute the statistics on these tracks and hypotheses when the local quantities may contain common information due to past communication.

We have developed an abstract model of the DSN in terms of the information graph. Using this graph, algorithms for information fusion have been developed. The two problems of hypothesis formation and evaluation all require keeping around histories of the tracks and hypotheses in the system. Using this history, the fusability of tracks and hypotheses can be determined. At the same time, any common information shared by their statistics can be identified so that it would not be double-counted. When specialized to broadcast communication, we can show that the general fusion algorithms for arbitrary communication reduce to those developed in the previous project.

The hypothesis formation algorithms for fusion do not depend on the target models. For hypothesis evaluation, we have assumed that the targets are static or that their motions may be approximated by "deterministic" process models. When the target models are assumed to be general Markov processes, the hypothesis evaluation algorithms have the same form as in (2.32) to (2.37). However, the state of a track would have to be a trajectory sampled at various times and computing its probability distribution would be difficult. Thus the difficulty of extending the results to treat general Markov models is more related to implementation issues. On the other hand, as long as the target motion is fairly regular, the deterministic process models we have assumed may be quite adequate.

3. DISSIMILAR SENSORS AND ATTRIBUTE BASED TRACKING

The algorithms presented in Section 2 are quite general and apply to arbitrary target models as long as the target motions are independent. In this section, we consider the case when the different nodes in the DSN have sensors of different types. For example, one node may have sensors which observe a certain set of features while another nodes may have sensors which observe a different set of features (e.g., radar versus acoustic). In general, the sensor produces data which contain attribute information as well as kinematic information. Typical attributes may include wheel or tread type of ground vehicles, radar images of ships, engine type of aircraft, and different types of electronic emissions.

This tracking problem with dissimilar sensors is both interesting from a theoretical and practical point of view since correlation of results from multiple sensors can often yield useful information not available from a single sensor. In particular, by considering attributes from multiple sensors, it may be possible to determine the type of the target.

In this section, we consider the problem of tracking and classifying targets when the nodes in the DSN have sensors of different types. Such targets usually have states which contain some structural information (e.g., a given target type may contain certain features which in turn contain other subfeatures). The relationship between targets with structured states and general structured set of targets will be discussed in Section 4.

3.1 TARGET AND SENSOR MODELS

We assume that the sensors at various DSN nodes have different capabilities and in particular, no single node can classify the target type uniquely. Thus, the nodes have to cooperate to achieve the overall mission. If this is not the case, then the results of Section 2 apply. Each node performs its local tracking and classification. Cooperation among the nodes, while it may improve the quality of the results, is not really necessary. There are also other situations when the nodes have identical sensors (e.g., acoustic or infrared) but the targets are not observable from a single sensor. In this case, cooperation among nodes is also needed. An example of this acoustic tracking will be presented in Section 5.

3.1.1 Target Models

We assume that each target has a state $x(t)$ at time t which evolves according to some dynamical model. The state $x(t)$ is represented by $(x^c(t), x^d(t))$, where $x^c(t)$ is the continuous part representing its geolocation quantities such as position, velocity, etc., while $x^d(t)$ is the discrete part representing other attributes. $x^c(t)$ is usually modeled by means of dynamical equations such as:

$$\dot{x}^c(t) = Fx^c(t) + w(t) \tag{3.1}$$

where $w(t)$ is a white driving noise. The components of x^d usually have some internal structure. For example, x^d may consist of:

$$x^d = (x^{d0}, x^{d1}, x^{d2}) \tag{3.2}$$

where

- x^{d0} is the type of the vehicle (tank or armored personnel carrier (APC), or truck)

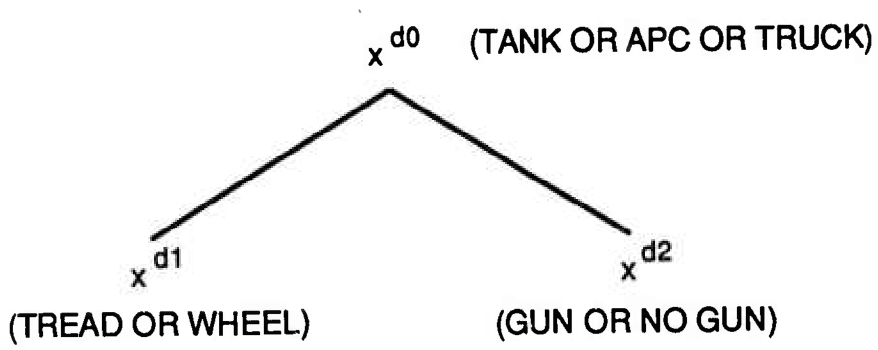
- x^{d1} is the attribute corresponding to the wheel type (tread or wheel)

- x^{d2} is the attribute corresponding to the weapon carried on the vehicle (gun or no gun)

The discrete states are related as in Figure 3-1 where the state x^{d0} determines the states x^{d1} and x^{d2}, i.e., the type of the vehicle determines the wheel type and the presence (absence) of guns as in Figure 3-1. In some cases, the relationship between the discrete states may also be probabilistic as given by $p(x^{d1}, x^{d2} \mid x^{d0})$. For example, one type of vehicle may have a given radio with certain probability. The probability of the attributes is sometimes conditionally independent, i.e.,

$$p(x^{d1}, x^{d2} \mid x^{d0}) = p(x^{d1} \mid x^{d0}) p(x^{d2} \mid x^{d0}) \tag{3.3}$$

which may simplify the processing considerably. In other cases, some discrete states themselves (other than the target type) may evolve with time and depend on other states, e.g, the electronic emission of a target. The dynamic behavior may be modeled by a Markov process.



TYPE ATTRIBUTE	TANK	APC	TRUCK
WHEEL TYPE	TREAD	TREAD	WHEEL
GUN TYPE	GUN	NO GUN	NO GUN

Figure 3-1: Example of Structured State

In general, the discrete target state may be hierarchical with more than two levels as given in Figure 3-2. Each attribute may assume different values depending on the target type. Frequently, the probability distribution of the attributes satisfy some Markov property, i.e., the probability of the attributes conditioned on all higher level attributes is the same as that conditioned on the attribute immediately above it. For the example in Figure 3-2, this implies that

$$p\,(x^{d\,31},x^{d\,32} \mid x^{d\,21},x^{d\,11},x^{d\,0}) = p\,(x^{d\,31},x^{d\,32} \mid x^{d\,21}) \tag{3.4}$$

3.1.2 Sensor Models

The sensors at the different DSN nodes may have different capabilities. Some sensors may measure the kinematic quantities while others may measure attributes (e.g., the wheel type or the absence or presence of guns). Still others may measure the target type directly. The sensors are subject to false alarms and mis-detections. For a detected target, the measurement model is given by $p_m(y_j \mid x)$ where x is the target state and y_j is the measurement for sensor j. To represent the presence of both kinematic and attribute measurements, the measurement model can be stated as

$$y_j^c(t) = H_j\, x^c(t) + v_j(t) \tag{3.5}$$

$$p\,(y_j^d(t) \mid x(t)) = p\,(y_j^d(t) \mid x^{di}, i \in A_j) \tag{3.6}$$

where

- $y_j^c(t)$ and $y_j^d(t)$ are the continuous and discrete components of the measurement of sensor j
- A_j is the set of attributes observable by sensor j
- $v_j(t)$ is the measurement noise

In the above, we have assumed that the continuous and discrete measurement models are independent. Sometimes this may not be the case; for example, a poor kinematic measurement may be correlated with a poor attribute measurement. The coupled measurement model then has to be used.

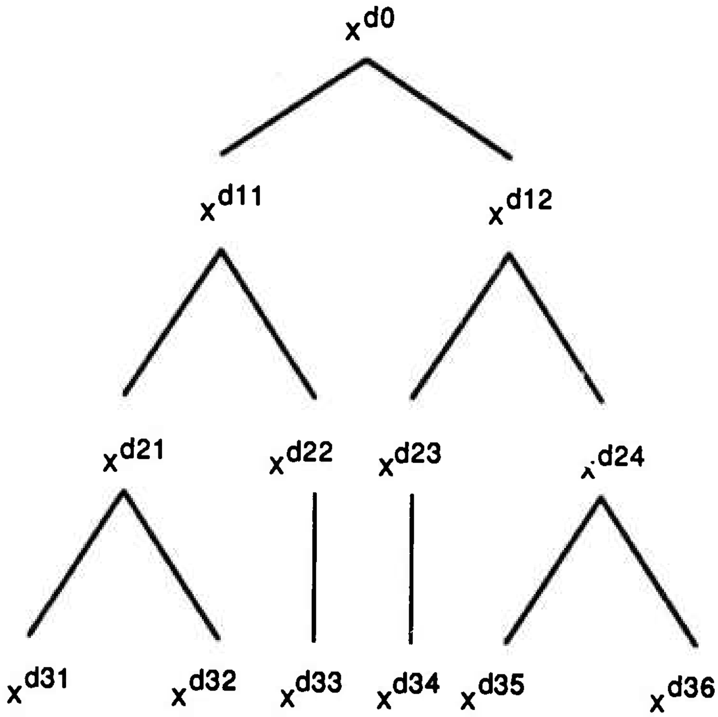


Figure 3-2: Hierarchical Target State

3.2 LOCAL PROCESSING

Local processing follows the algorithms presented in Section 2.1.1, using the appropriate sensor model for each node. Each node has an information state represented by the set of tracks, set of hypotheses, track state distributions and hypothesis probabilities. Equation (2.2) is used for hypothesis evaluation with the likelihoods given by (2.3) to (2.5). Since not all the states of the targets are observable from the sensor j, the state x should be replaced by:

$$x^j = (x^c, x^{dj}) \tag{3.7}$$

where x^c is the continuous (geolocation) state and x^{dj} is the discrete state of attributes observable from sensor j. The relevant tracks state distribution is then (with some independence assumptions on the states and measurements)

$$p_t(x^j \mid \tau, Z_j) = p(x^c \mid \tau, Z_j) p(x^{dj} \mid \tau, Z_j) \tag{3.8}$$

When the target and sensor models for the continuous state satisfy linear and Gaussian models, the geolocation component of the track state description can be computed by means of the Kalman Filter. The discrete component is computed using a Bayesian updating formula. Assuming x^{dj} is static, then

$$p(x^{dj} \mid \tau, Z_j) = C^{-1} p_m(y_j^d \mid x^{dj}) p(x^{dj} \mid \bar{\tau}, \bar{Z}_j) \tag{3.9}$$

where C is a normalization constant, $p_m(\cdot)$ is the discrete measurement model and $p(x^{dj} \mid \bar{\tau}, \bar{Z}_j)$ is the predicted discrete state given the previous measurements.

For the example of Figure 3-1, each hypothesis from the wheel type sensor node will contain the number of targets detected, their positions and velocities and possible classifications into wheel and tread vehicles (with probabilities). Similarly, the gun type sensor generates hypotheses with tracks described by gun type as well as locations and velocities.

3.3 INFORMATION FUSION

When sensor nodes have the same type of sensors, communication among nodes serves primarily to reduce the uncertainty associated with the situation assessment at each node. For example, nodes 1 and 2 may have different estimates of a target given by $\hat{x}_i^c, \Sigma_i, p_i(x^d)$), $i = 1,2$, where

- $\hat{x}_i^c$ is the geolocation estimate by sensor node i

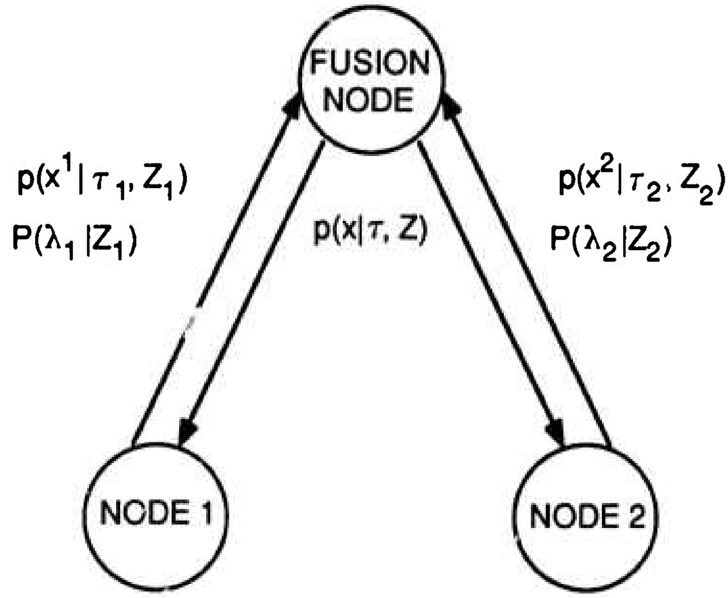
- Σ_i is the error covariance of sensor node i, and

- $p_i(x^d)$ is the probability distribution of the discrete state x^d estimated by sensor i.

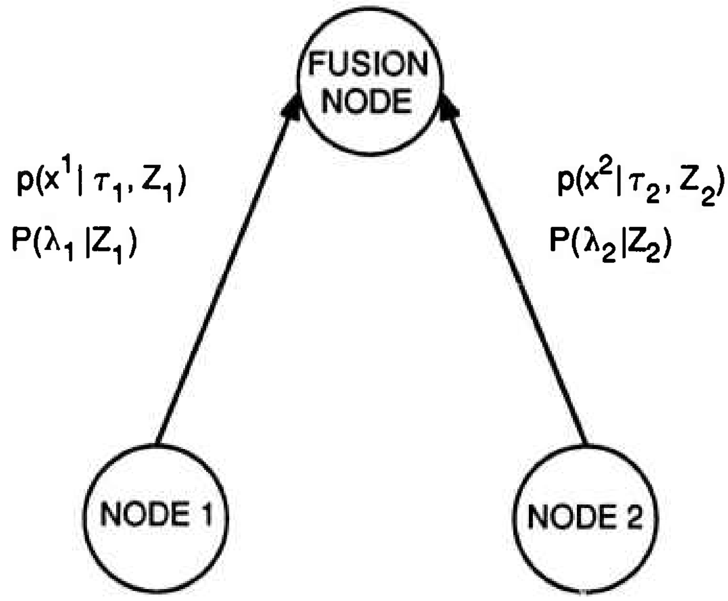
Then when the nodes communicate, the estimate of the target can be improved and becomes $(\hat{x}^c, \Sigma, p(x^d))$ through fusion of the track state estimates. In addition, the nodes can also improve on their estimates of the number of targets.

When the nodes have sensors of different types, each node produces track state estimates for the attributes which are observable to the node. Communication between nodes then not only improves the geolocation estimates but also produces estimates of other attributes not observable from the individual nodes. This will usually require knowledge of the relationship among the attributes in the structured state. For example, if Node 1 concludes that the target is a vehicle with thread and Node 2 concludes that it has a gun, then through communication each node may conclude that it is a tank. If the individual nodes' estimates are probabilistic, then the fusion results will also be probabilistic.

In the following, we consider information fusion for nodes with dissimilar sensors. Hypothesis formation and management follow the general algorithm given in Section 2. For example, fusability conditions will have to be checked before tracks and hypotheses are fused. Our discussion will thus focus on fusion of track state estimates and hypothesis evaluation. Since the fusion results for arbitrary communication can be derived from fusion of two nodes, we assume the structures in Figure 3-3. In Figure 3-3(a), the fusion node can be a different node from nodes 1 and 2. It collects information from Node 1 and Node 2, performs fusion and broadcasts the results back to the nodes, thereby performing the coordination. Alternatively, the fusion node may reside with each of the two nodes in a broadcast situation. Figure 3-3(b) is the case with no coordination or feedback from the fusion node to Nodes 1 and 2.



(a) With Coordination



(b) Without Coordination

Figure 3-3: Communication Structure

3.3.1 Fusion of Track State Estimates

Since fusion of track state estimates is for tracks which have been associated, we do not represent the track explicitly in the following discussion. Suppose the observable state for Node i is x^{i} which includes the continuous state x^{c} and the discrete state x^{di}, i.e.,

$$x^{i} = (x^{c}, x^{di}) \tag{3.11}$$

For a given target, the track state estimate by Node i given the cumulative data Z_i is given by $p(x^{i} \mid Z_i)$. As discussed before, this may contain a continuous part (mean and covariance) and a discrete part (probability distribution). For the example of Figure 3-1,

$$p(x^{i} \mid Z_i) = p(x^{c}, x^{di} \mid Z_i) = p(x^{c} \mid Z_i) p(x^{di} \mid Z_i) \tag{3.12}$$

where $p(x^{c} \mid Z_i)$ will be characterized by a mean and covariance. Let $\bar{Z}$ be the cumulative data of the fusion node when it last broadcast and Z be the cumulative data after it receives communication from the nodes, then the complete state estimate of the target after fusion is given by:

$$p(x^{c}, x^{d0}, x^{d1}, x^{d2} \mid Z) = p(x^{c} \mid Z) p(x^{d0}, x^{d1}, x^{d2} \mid Z) \tag{3.13}$$

where the continuous and discrete state estimates are computed as below.

Fusion with Coordination

The fusion of the continuous state estimate is given by

$$p(x^{c} \mid Z) = C_1^{-1} \frac{p(x^{c} \mid Z_1) p(x^{c} \mid Z_2)}{p(x^{c} \mid \bar{Z})} \tag{3.14}$$

where C_1 is a normalization constant. The fusion of the discrete state estimate is given by

$$p(x^{d0}, x^{d1}, x^{d2} \mid Z) = C_2^{-1} \frac{p(x^{d1} \mid Z_1) p(x^{d2} \mid Z_2)}{p(x^{d1} \mid \bar{Z}) p(x^{d2} \mid \bar{Z})} p(x^{d0}, x^{d1}, x^{d2} \mid \bar{Z})$$

$$= C_2^{-1} \frac{p(x^{d1} \mid Z_1) p(x^{d2} \mid Z_2)}{p(x^{d1} \mid \bar{Z}) p(x^{d2} \mid \bar{Z})} \cdot p(x^{d1}, x^{d2} \mid x^{d0}) p(x^{d0} \mid \bar{Z}) \tag{3.15}$$

where we have assumed that the attributes x^{d1} and x^{d2} depend only on x^{d0} and C_2 is a normalization constant.

Equation (3.14) is the standard equation for fusing two probability distributions of the same random state. Equation (3.15) fuses the probability distributions of different attributes to obtain that of all attributes. The last factor in (3.15) represents the a priori estimate of the attribute x^{d0} (vehicle type) based on $\bar{Z}$. It is the marginal probability of $p(x^{d0}, x^{d1}, x^{d2} \mid \bar{Z})$ computed from an earlier fusion. The factor $p(x^{d1}, x^{d2} \mid x^{d0})$ is the model of the structured state. For the example in Figure 3-1, we may have

$$p(x^{d1} = \text{tread}, x^{d2} = \text{gun} \mid x^{d0} = \text{tank}) = 1$$

$$p(x^{d1} = \text{tread}, x^{d2} = \text{no gun} \mid x^{d0} = \text{APC}) = 1$$

$$p(x^{d1} = \text{wheel}, x^{d2} = \text{no gun} \mid x^{d0} = \text{truck}) = 1 \qquad (3.16)$$

and zero otherwise. The last two factors in equation (3.15) together predict the target attributes from the previous communication time. The first two factors in the numerator represent the estimates of the two attributes from the two nodes. Since these estimates share some information with the predicted values through $\bar{Z}$, the factors in the denominator are used to remove any redundant information. To obtain an estimate of the target type, one needs only to sum over the possible values of x^{d1} and x^{d2} to obtain $p(x^{d0} \mid Z)$.

Fusion Without Coordination

If there is no feedback from the fusion node to the other nodes, then Equation (3.14) should be replaced by

$$p(x^c \mid Z) = C_1^{-1} \frac{p(x^c \mid Z_1) p(x^c \mid Z_2)}{p(x^c \mid \bar{Z}_1) p(x^c \mid \bar{Z}_2)} p(x^c \mid \bar{Z}) \qquad (3.17)$$

where C_1 is a normalization constant, $\bar{Z}_i$ is the cumulative data of Node i before the last communication to the fusion node. This fusion formula is a special case of the general Equation (2.17).

Similarly, the fusion formula for the discrete state estimate is given by

$$p(x^{d0},x^{d1},x^{d2} \mid Z) = C_2^{-1} \frac{p(x^{d1} \mid Z_1)p(x^{d2} \mid Z_2)}{p(x^{d1} \mid \bar{Z}_1)p(x^{d2} \mid \bar{Z}_2)} \tag{3.18}$$

$$p(x^{d1},x^{d2} \mid x^{d0})p(x^{d0} \mid \bar{Z})$$

where C_2 is another normalization constant. This equation is similar to (3.15) except for the terms in the denominator which now depend on $\bar{Z}_1$ and $\bar{Z}_2$ instead of $\bar{Z}$. The reason for using the different terms can be seen by drawing the information graphs.

So far we have assumed that the fusion node does not have any measurements of its own. If this is not the case, as when the sensor measures the type x^{d0} discrete, then the fusion equations can be modified appropriately. The distributed hierarchical Bayesian Approach of [5] can also be used to estimate the attributes in the target state. Although such an approach also has a distributed implementation, it is not as convenient as the approach used here if data association also has to be considered.

3.3.2 Hypothesis Evaluation

In the previous section, we have considered the fusion of state (both continuous and discrete) estimates for the individual target tracks. In multitarget tracking, the main problem is data association and track association in the case of multiple sensor nodes. Thus, we need to evaluate the probability of each track-to-track association hypothesis. The general hypothesis evaluation formula (Equation (2.32)) is applicable to this special case. We again illustrate the algorithm with the example of Figure 3-1.

Fusion with Coordination

The hypothesis evaluation equation is

$$P(\lambda \mid Z) = C^{-1} \frac{P(\lambda_1 \mid Z_1)P(\lambda_2 \mid Z_2)}{P(\bar{\lambda} \mid \bar{Z})} \prod_{\tau \epsilon \lambda} l_\tau \tag{3.19}$$

where the probabilities $P(\lambda_i \mid Z_i)$, $\lambda = 1, 2$, are communicated from nodes 1 and 2 to the fusion node, $P(\bar{\lambda} \mid \bar{z})$ is the a priori probability of the predecessor hypothesis at the fusion node, and C is a normalization constant.

The track-to-track association likelihood L_τ can be derived from Equation (2.33) and is given by

$$l_\tau = \int \frac{\tilde{p}(x \mid \tau_1, Z_1)\tilde{p}(x \mid \tau_2, Z_2)}{\tilde{p}(x \mid \overline{\tau}, \overline{Z})} \mu(dx) \tag{3.20}$$

where τ_1 and τ_2 are tracks where are fused t[illegible]n τ and $\tilde{p}(\cdot)$ is as defined in Equation (2.35). The likelihood l_τ can be further decomposed into two likelihoods

$$l_\tau = l_\tau^c \; l_\tau^d \tag{3.21}$$

where l_τ^c is the likelihood computed from the continuous state and l_τ^d is computed from the discrete state. The continuous likelihood is given by:

$$l_\tau^c = \int \frac{\tilde{p}(x^c \mid \tau_1, Z_1)\tilde{p}(x^c \mid \tau_2, Z_2)}{\tilde{p}(x^c \mid \overline{\tau}, \overline{Z})} dx^c \tag{3.22}$$

and depends on how close the geolocation state estimates of the two tracks are. The discrete likelihood is given by

$$l_\tau^d = \sum_{x^{d0}, x^{d1}, x^{d2}} \frac{p(x^{d1} \mid \tau_1, Z_1) p(x^{d2} \mid \tau_2, Z_2)}{p(x^{d1} \mid \overline{\tau}, \overline{Z}) p(x^{d2} \mid \overline{\tau}, \overline{Z})} \; p(x^{d1}, x^{d2} \mid x^{d0}) p(x^{d0} \mid \overline{\tau}, \overline{Z}) \tag{3.23}$$

and depends on how well the attribute estimates from the two nodes match the prior estimate of the target type according to the model of the structured state.

Note that the likelihood computation is closely coupled to the fusion of the track state estimates from the similarity between Equations (3.14) and (3.22), and between (3.15) and (3.23). In fact, the normalization constants in Equations (3.14) and (3.15) are the likelihoods. Thus, likelihood computation and track state fusion are usually performed at the same time.

Fusion Without Coordination

In the case where there is no feedback from the fusion node, the hypothesis evaluation equation is given by

$$P(\lambda \mid Z) = C^{-1} \frac{P(\lambda_1 \mid Z_1) P(\lambda_2 \mid Z_2)}{P(\overline{\lambda}_1 \mid \overline{Z}_1) P(\overline{\lambda}_2 \mid \overline{Z}_2)} P(\overline{\lambda} \mid \overline{Z}) \prod_{\tau \in \lambda} l_\tau \tag{3.24}$$

where $P(\lambda_i \mid Z_i)$ is defined as before, $\bar{\lambda}_i$ is the restriction of the hypothesis λ_i to $\bar{Z}_i$ and $P(\bar{\lambda}_i \mid \bar{Z}_i)$ is the probability of the hypothesis $\bar{\lambda}_i$ before communication to the fusion node.

The track-to-track association likelihood l_τ again can be decomposed into:

$$l_\tau = l_\tau^c \; l_\tau^d \tag{3.25}$$

where the continuous likelihood is given by

$$l_\tau^c = \int \frac{p(x^c \mid \tau_1, Z_1) p(x^c \mid \bar{\tau}_1, Z_2)}{p(x^c \mid \bar{\tau}, \bar{Z}_1) p(x^c \mid \bar{\tau}, \bar{Z}_2)} \, p(x^c \mid \bar{\tau}, \bar{Z}) dx^c \tag{3.26}$$

and the discrete likelihood is given by

$$l_\tau^d = \sum_{x^{d0}, x^{d1}, x^{d2}} \frac{p(x^{d1} \mid \tau_1, Z_1) p(x^{d2} \mid \tau_2, Z_2)}{p(x^{d1} \mid \bar{\tau}_1, \bar{Z}_1) p(x^{d2} \mid \bar{\tau}_2, \bar{Z}_2)} \tag{3.27}$$

$$p(x^{d1}, x^{d2} \mid x^{d0}) p(x^{d0} \mid \bar{Z})$$

As in the fusion with coordination case, the likelihood computation and track state fusion operations can be performed together. Note the similarity of these equations to those used for fusion with coordination. Equations (3.23) and (3.27) are almost the same except for the conditioning of the information in the denominator. In both cases, the structural relationship between the discrete states is used to evaluate the likelihood of association.

3.4 CONCLUSION

In this section, we have applied the general results of Section 2 to the case of nodes with dissimilar sensors. When an individual node can only observe certain attributes of the target state, cooperation among nodes observing different attributes can improve the performance of the system significantly. By exploiting knowledge on the structured state, the receiving node can assess some missing attributes such as the target type.

We have presented the fusion results via a specific example and communication structure. The algorithms for handling more general cases cn be developed along the same principles. The fusion algorithms consist of two closely coupled operations: fusion of target state estimates and evaluation of association likelihoods. For the continuous states (position, velocity) these operations are the same as the case of similar sensors. The discrete states, however, involve operations which make use of the knowledge of the structured state. Therefore, there is no theoretical difficulty to treat targets with structured states and dissimilar sensors with measurement at different levels. In practice, however, classification trees for the discrete attributes may be very complicated and the number of terminal nodes may be simply too many to handle in a straightforward way. In such a case, we need additional tools to effectively store and update the probability distributions on the entire terminal nodes. In [6] and [7], a set of procedures to solve such problems is shown by means of an example of ocean surveillance. Many of hypothesis management procedures devised for controlling data-to-data hypotheses (e.g., those described in [6] and being developed in the current project) can be extended to provide useful tools, e.g., hypothesis pruning, hypothesis combining and clustering. Furthermore, effective representation of probability distributions must be developed in order for such management systems to work effectively. For example, track state distributions of tracks may have different representations depending on their status. Distributed processing on different levels may also be an effective procedure. Some of the results in [5] may also be applicable.

4. TRACKING AND CLASSIFYING STRUCTURED TARGETS

By structured targets, we may mean two different concepts in multitarget tracking:

(1) targets with structured states
(2) structured sets of targets.

Since each individual target may be represented on an individual target space, concept 2 is one-level higher than 1. In a model based on the above concept 1, targets are still treated as individual objects although correlation among them can be considered and targets may be governed by a common state as a group of targets. This kind of models is necessary, when a multilevel identification process for each target is used or when a target has structured features. Such issues are related to the problem of treating dissimilar sensors which generate measurements corresponding to different levels of the structured target state space. This problem has been discussed in Section 3.

On the other hand, concept 2 is essential when targets are in fact organized and structured in units at various levels. A typical example can be found in military units such as army → division → regiment → battalion → company, etc., in the military hierarchy. In such a case, the number of targets is typically very large and, if they are treated as independent objects, we may not be able to assess a global situation based on the outputs from any reasonably functioning target tracking system. This is so because, since grouped targets are usually closely spaced, the data-to-data association (or scan-to-scan correlation) may become very difficult with limited computational resources. This difficulty may be overcome only when the unit structure of targets is understood and taken into account in a tracking system. Moreover, the global assessment of all the targets as a single structured object is itself an important task in many applications.

Our emphasis has been the development of a general theory upon which we may produce effective algorithms in many different applications. This should serve also as a basis for developing distributed algorithms. Section 4.1 discusses a general model for structured set of targets. In Section 4.2, we will present our first-cut analysis on structured sets of targets. An algorithm is derived for two-level structured targets, i.e, tracking groups of targets. The future direction of our algorithm development effort will be discussed in the concluding Section 4.3.

4.1 MODEL FOR STRUCTURED SETS OF TARGETS

A typical example of a structured set of targets is shown in Figure 4-1 in which a division in an army is shown in a simplified way. Depending on the type of the division, the composition and the number of subordinates, i.e., battalions have a certain pattern. The same kind of dependence is also present in the relationship between the subsequently lower levels. This kind of structure produces another dimension to the multitarget tracking problems. There are only very few theoretical results on tracking and classification of structured sets of targets. Besides a few documents referred in [8], we can only refer to a couple of technical references, [9] and [10], both of which are concerned with two-level tracking, i.e., tracking of groups of targets, but treat issues pertaining multiple groups in a rather ambiguous manner. On the other hand, AI (Artificial Intelligence) -type approaches were used in much more complicated environments in [11] and [12] which are concerned with ocean surveillance and battlefield unit identification, respectively. [11] uses a single-hypothesis propagation combined with a backtracking-like recovery scheme while [12] adopts a multi-hypothesis approach. The systems described in [11] and [12] may be viewed as hierarchical systems which may be illustrated as in Figure 4-2. The procedures represented by upward arrows are often called *bottom-up* or *induction* processes and those represented by downward arrows *top-down* or *deduction* processes.

While the decomposition illustrated by Figure 4-2 is certainly a key to successful implementation of the systems described in [11] and [12], each hypothesis evaluation cannot be performed independently in general. For example, in tracking groups of targets, we must hypothesize possible group formation from input data while, at the same time, the states as a group must be determined and then the estimation of the states affects the evaluation of lower level hypotheses. Even if the bottom-up/top-down updating is clearly defined, iterations may be necessary for such processes to converge. Moreover, in some cases, a simple bottom-up type process may easily be overwhelmed by combinatorics. Therefore, at least for the few lower levels, we may need an integrated approach rather than a decomposition approach taken in [11] and [12]. In the subsequent subsections, we will try to establish a first-cut analysis which treats the whole structure of targets in an integrated manner.

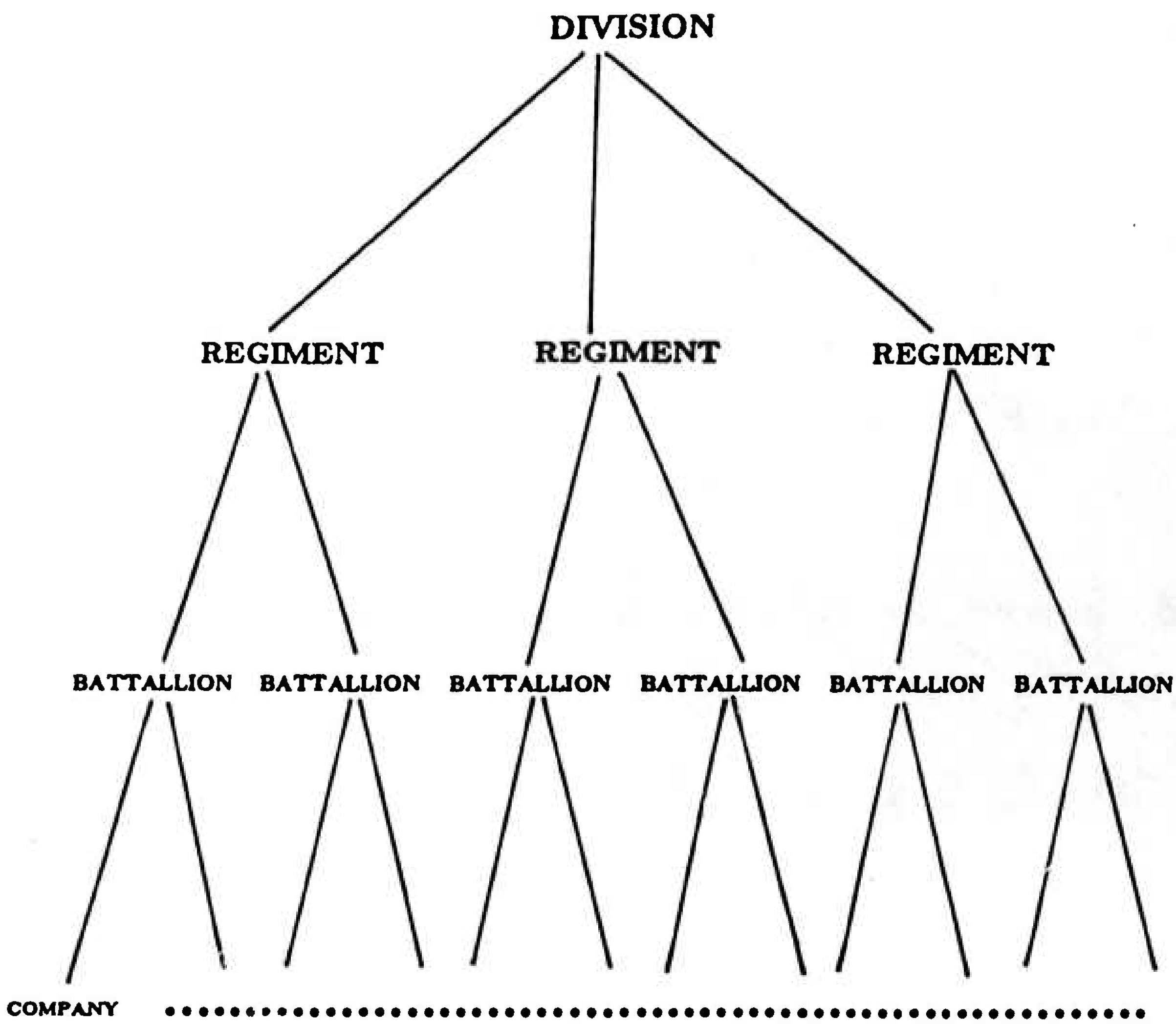


Figure 4-1: Structured Sets of Targets

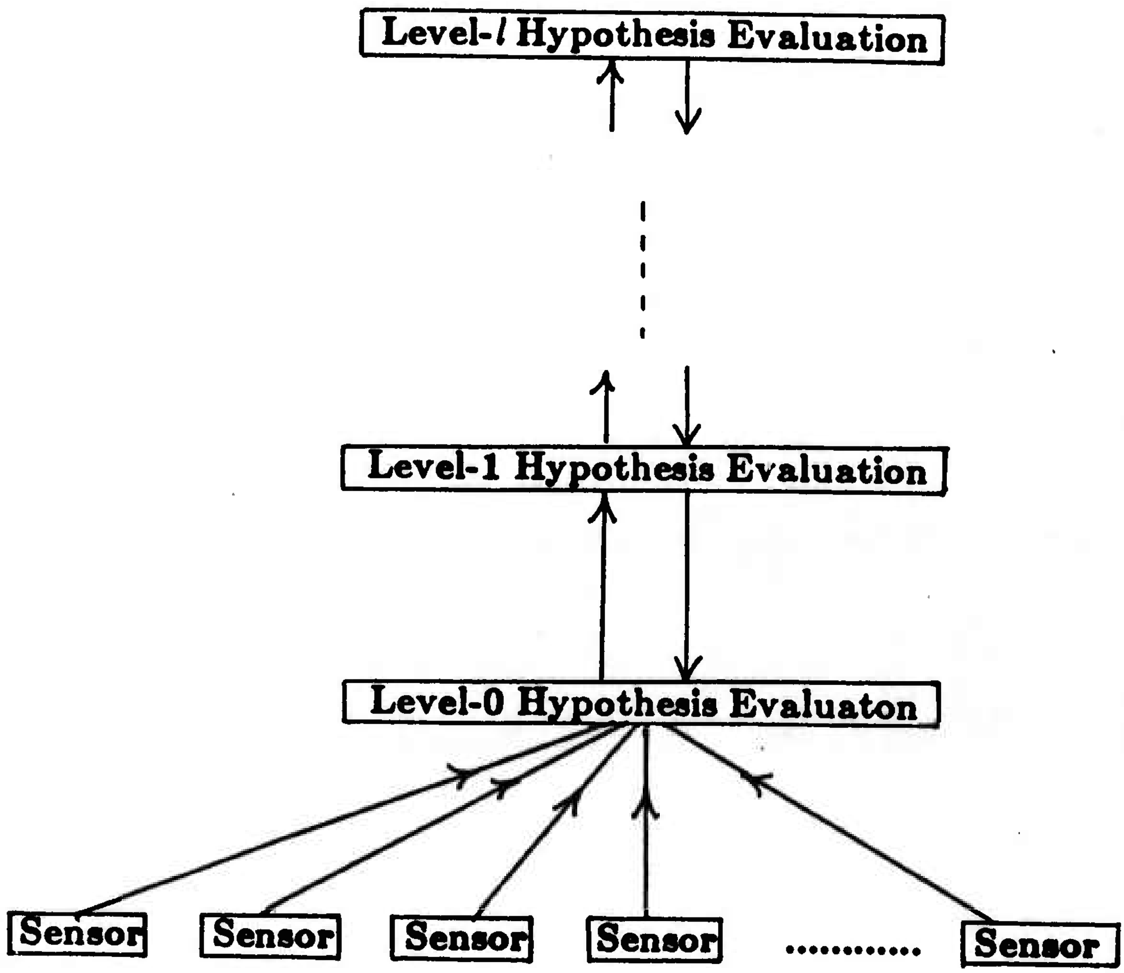


Figure 4-2: Hierarchical Hypothesis Evaluation

4.1.1 A Model for Structured Sets of Targets

When we focus on each node in Figure 4-1 and its immediate successors rather than the whole picture, we notice the tree is composed of building blocks each of which has the same structure. Such a building block can be identified with a structure of a state representing a group of targets, as shown in Figure 4-3a. In tracking and classifying a group of targets, the totality of targets can be represented by (1) [level 1] the total number of targets plus a common target state component for the group, and (2) [level 0] the states of individual targets. (1) is one-level higher than (2) since (2) cannot be defined unless the number of targets is given by (1). This structure can be extended to the cases where multiple groups of targets are present. Such a case may be represented by a tree which may be illustrated in Figure 4-3b. Each level of nodes in Figure 4-3b represents: (1) [level 2] the total number of groups plus a common state component for all the groups, (2) [level 1] the states of individual groups including, for each group, the number of targets in the group and a common state component for all the targets in the group, and (3) [level 0] the states of individual targets in each group.

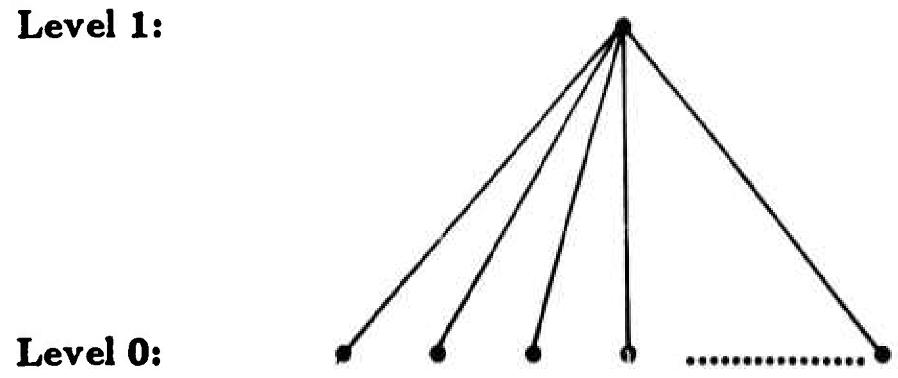
This approach can be extended to an arbitrary level l of structures. We call such a structure a *level-l target structure* or simply a *level-l target*. As seen in Figure 4-3, when a tree represents a level-l target structure, the nodes in the tree can be labeled as level 0, level 1, ..., level l. There is always only one node at the highest level, i.e., level l. The nodes at the lowest level, i.e., level 0, represents the set of all the *targets* which we may call *level-0 targets*. In a formal description, we define a *level-l' state* for a *level-l' target* i as

$$x_i^{(l')} = (N_i^{(l')}, x_{i0}^{(l')}, x_{i1}^{(l'-1)}, \ldots\ldots, x_{iN_i^{(l')}}^{(l'-1)}) \tag{4.1}$$

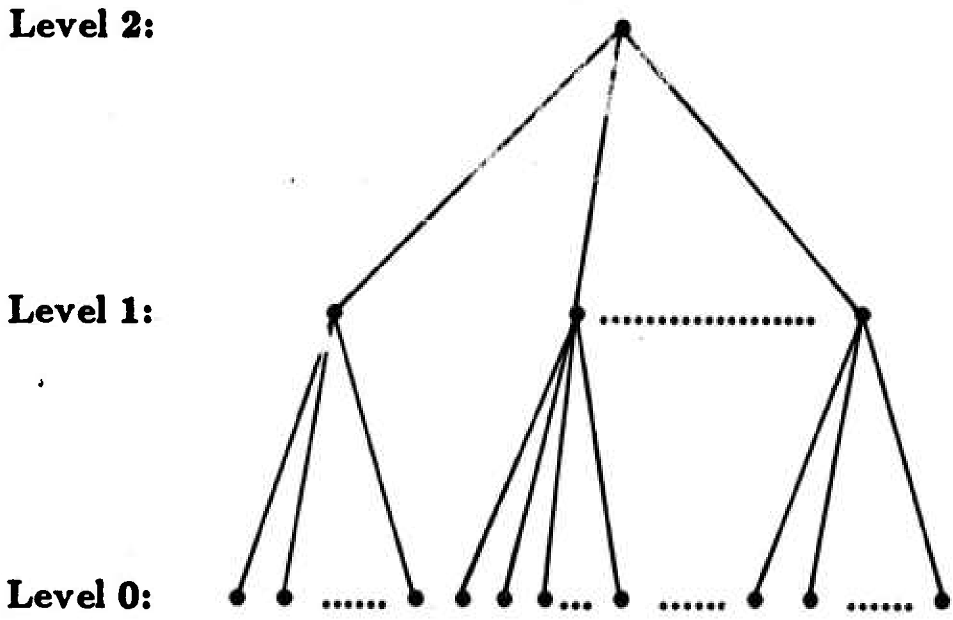
where $N_i^{(l')}$ is the number of the *level*-$(l'-1)$ targets in the level-l' target i, $x_{i0}^{(l')}$ is the state component common to all the level-$(l'-1)$ targets contained in level-l' target i, and each $x_{ij}^{(l'-1)}$ is the state of the j-th level-$(l'-1)$ target. Unless $l'=1$ in (4.1), every $x_{ij}^{(l'-1)}$ is defined similarly with l' being replaced by $l'-1$. When $l'=l$, there is no need to use index i in (4.1). Each level-l' target when $l'<l$ is therefore indexed as

$$i = (i_{l-1}, \ldots\ldots, i_{l'}) \ . \tag{4.2}$$

According to an alternative view of this approach, we are first given a set of targets, then a partition of the targets into multiple groups, then a partition of



(a) Single-Level Targets



(b) Two-Level Targets

Figure 4-3: Single-Level and Two-Level Targets the groups into multiple super-groups, and so forth. In other words, a level-l' target is an element of a partition of the set of all the level-(l'–1) targets. The partition is a trivial one when $l'=l$. In typical battlefield units as shown in Figure 4-4, each unit has its headquarter (division command post (DCP), regiment headquarter (RH), battalion headquarter (BH), etc.) besides its subordinates (R = a regiment, B = a battalion, C = a company, etc.). These headquarters may be considered either (1) as a part of the common state of each level-l' target or (2) as special targets which do not have any subordinate. When we adopt the latter consideration, we may simply extend each headquarter node to the lowest level, i.e., level 0. As mentioned before, as a first-cut analysis, we ignore such problems. There will be no problem in rectifying the formulation to treat headquarters in appropriate ways in the future.

4.1.2 Sensor Models and Multi-Level Tracks and Hypotheses

We can extend our target-sensor model for multitarget tracking from single-level cases to multi-level cases in a rather straightforward way as follows: Let S be a finite set of *sensors* which observe the targets. For each sensor s, the *measurement value space* Y_s in which measurements from sensor s take values is assumed to be a hybrid space. Each output from sensor s is a *data set* $(y_1, \ldots, y_m, m, t, s)$ which is an element of

$$\bigcup_{m=0}^{\infty} \bigcup_{s \in S} (Y_s)^m \times \{m\} \times [t_0, \infty) \times \{s\}$$

and represents m measurements, $y_1, \ldots\ldots, y_m$, generated by sensor s at time t. (t_0 is the time before which no sensor outputs any data set.) A collection of data sets available up to a certain time is called a *cumulative data set*. We assume that all the data sets are indexed by positive integers as $z(1), z(2), \ldots\ldots$, where

$$z(k) = ((y_j(k))_{j=1}^{N_M(k)}, N_M(k), t_k, s_k) \tag{4.3}$$

for each positive k such that $t_k \le t_{k'}$ whenever $k < k'$. A *cumulative measurement set up to k* is defined as

$$J^{(k)} = \bigcup_{k'=1}^{k} \{1, \ldots\ldots, N_M(k')\} \times \{k'\} \tag{4.4}$$

For the sake of simplicity, we assume that possible origins of measurements in any data set are only level-0 targets. Let I_T be the set of level-0 target indices. For each data set k, we assume an *assignment function* A_k defined on a

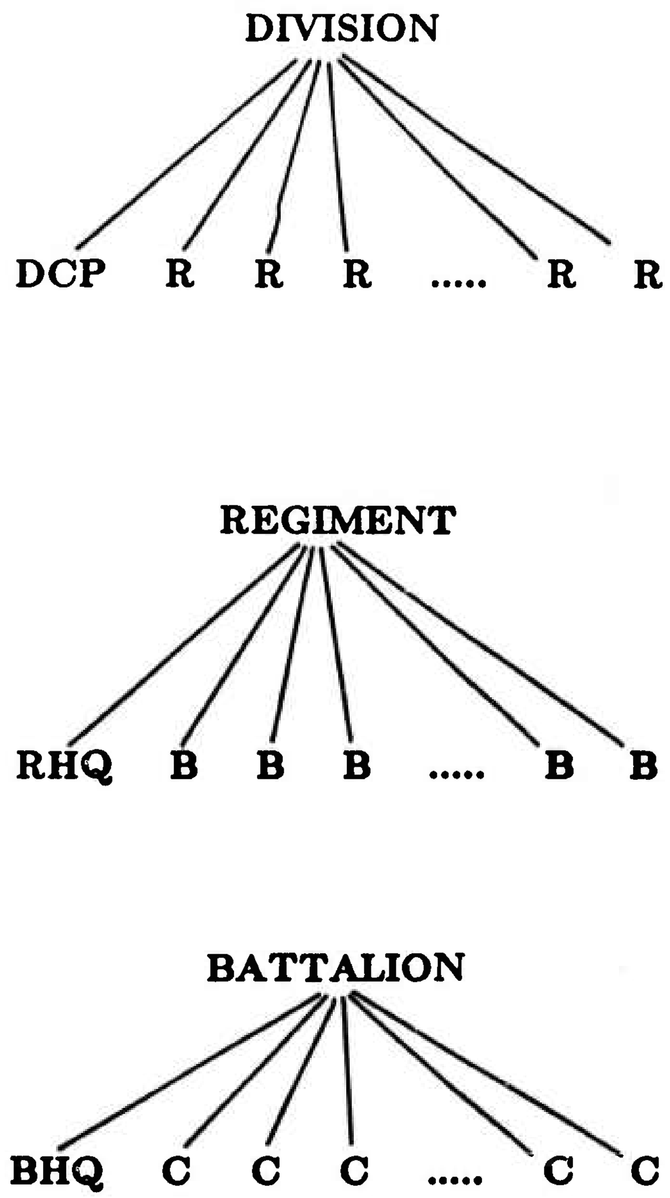


Figure 4-4: Battlefield Units

subset of the level-0 target index set I_T taking values in $J(k) \triangleq \{1,\ldots,N_M(k)\}$. When $j = A_k(i_{l-1},\ldots,i_0)$, we say level-0 target $(i_{l-1}, \ldots, i_0)$ is *detected by sensor s_k at time t_k and generates* the *j-th measurement, or the j-th measurement originates from* level-0 target $(i_{l-1}, \ldots, i_0)$. With the no-split/no-merged measurement assumption, such an A_k is a well-defined one-to-one function. Then, given a cumulative data set, we can define the trace of a level-l' target in it in the form of a subset of the cumulative measurement index or a collection of such subsets at the given level. We call any possible realization of such a trace a *level-l' track*. Thus a subset of the measurement index set is a level-0 track if it contains at most one measurement index set for each data set. A level-l' track is a collection of nonoverlapping level-$(l'-1)$ tracks. A *level-l' hypothesis* is then a collection of nonoverlapping nonempty level-l' tracks and hypothesizes all the set of measurements originating from level-l' targets. According to this definition, a level-1 track is also a level-0 hypothesis, and *vice versa*, although its interpretation as a track is completely different from that as a hypothesis.

Multi-level hypotheses defined above may be illustrated in Figure 4-5 in which $l=3$ and a level-2 hypothesis is represented by a tree depicted by solid lines. In Figure 4-5, the level-2 hypothesis consists of two level-2 tracks each of which hypothesizes a group of detected groups of targets, $\{\{\tau_1,\tau_2\},\{\tau_3,\tau_4\}\}$ $\{\{\tau_5\},\{\tau_6,\tau_7\}\}$, where τ_1 to τ_7 are level-0 tracks each of which hypothesize a detected level-0 target. Given such a hypothesis, we must further hypothesize the existence of undetected targets and the overall structure, as shown in Figure 4-5 by broken lines. The process to group given level-0 tracks τ_1 to τ_7 in a level-0 hypothesis into a level-1 hypothesis and then into a level-2 hypothesis can be viewed as a *bottom-up* procedure. While the process to add hidden targets and to complete the overall structure can be viewed as a *top-down* procedure. The evaluation of hypotheses may not be, however, decomposed in such a manner. The discussion of hypothesis evaluation in a general level-l case may be very complicated. Therefore, in the following sections, we will restrict ourselves to the cases where $l=2$, i.e., where tracking of multiple groups of targets is concerned.

Remark: In the above discussion, we only considered the cases where each measurement from each sensor is based on a level-0 target. The definitions of tracks and the hypotheses may be altered so that measurements from different levels may be treated. At this moment, however, the exact form of the appropriate modification is not very clear.

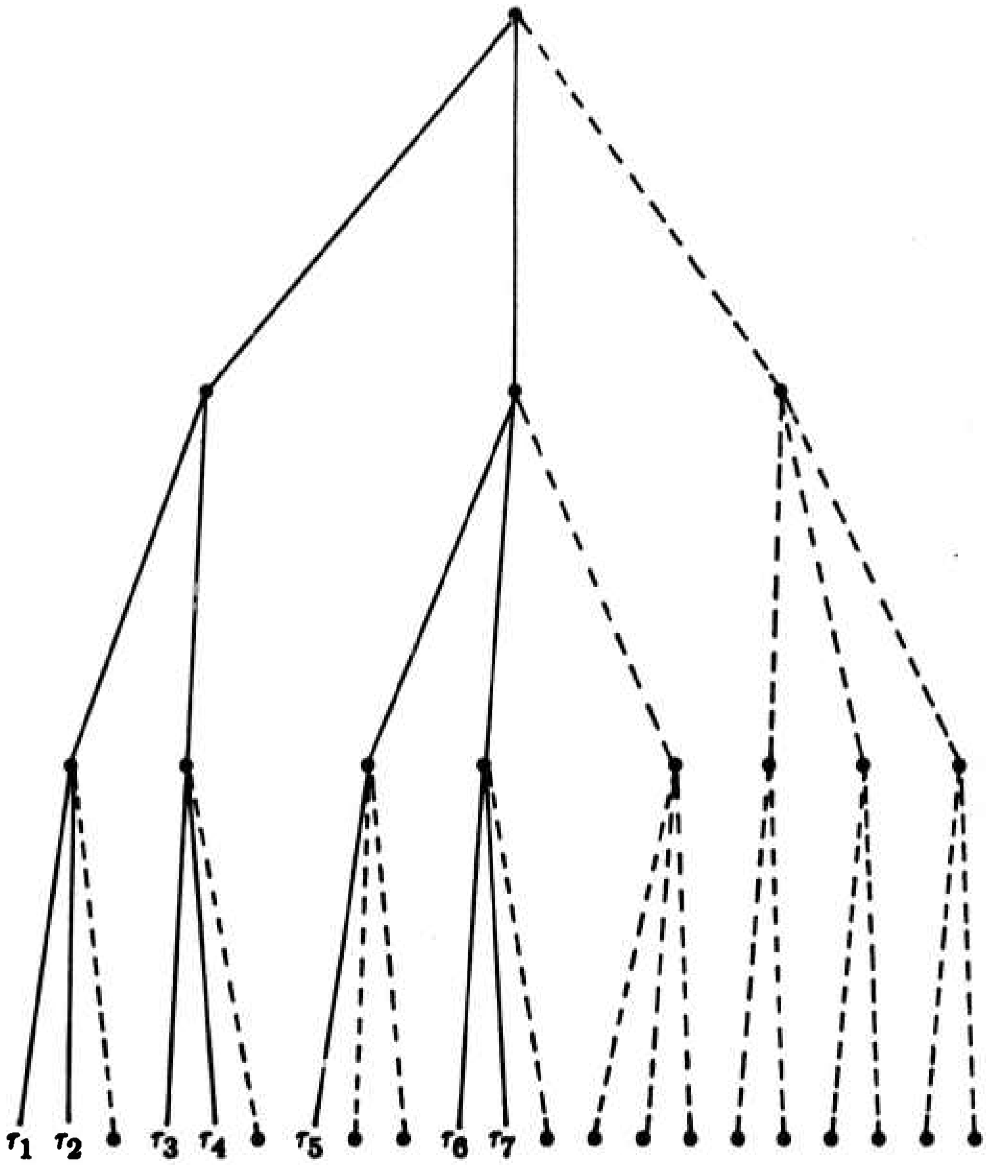


Figure 4-5: Multi-Level Hypothesis

4.2 EVALUATION OF TWO-LEVEL HYPOTHESES

In this subsection, we will extend our general theory of multitarget tracking from single-level cases to two-level cases, i.e., tracking multiple groups of targets. The issues pertaining to implementation will be briefly discussed in terms of an example.

4.2.1 Two-Lev Multitarget Tracking

When the target structure level is two, i.e., $l=2$, the overall target state can be written as

$$x = (N_G, x_0, x_1, \ldots, x_{N_G}) \tag{4.5}$$

where N_G is the total number of groups, x_0 is the state component common to all the groups, and each x_i is the i-th group's individual state. Each x_i is therefore in form of

$$x_i = (N_i, x_{i0}, x_{i1}, \ldots, x_{iN_i}) \tag{4.6}$$

where N_i is the number of (level-0) targets in group i, x_{i0} is the state component common to all the targets in group i, and x_{ij} is the individual states of the j-th target in group i. Let the level-1 target index set be $I_G=\{1,\ldots,N_G\}$ and the level-0 track index set be $\bigcup_{i=1}^{N_G}\{i\}\times\{1,\ldots,N_i\}$. Then the trace of level-0 target (i_1,i_0), i.e., the i_0-th target in the i_1-th group, in a cumulative data set up to k is

$$T_k^{(0)}(i_1,i_0) = \{(j,k') \mid j=A_{k'}(i_1,i_0), 1\le k'\le k\} \quad . \tag{4.7}$$

The trace of level-1 target i_1 is then

$$T_k^{(1)}(i_1) = \{T_k^{(0)}(i_1,i_0) \mid 1\le i_0\le N_{i_1}\} \quad . \tag{4.8}$$

Then a level-0 hypothesis is a possible realization of

$$\Lambda_k^{(0)} = \{T_k^{(0)}(i_1,i_0) \mid T_k^{(0)}(i_1,i_0)\neq\emptyset, (i_1,i_0)\in I_T\} \tag{4.9}$$

and a level-1 hypothesis is a possible realization of

$$\Lambda_k^{(1)} = \{T_k^{(1)}(i_1) \mid T_k^{(1)}(i_1)\neq\{\emptyset\}, i_1\in I_G\} \quad . \tag{4.10}$$

We can extend the concept of *target-to-track hypothesis* from single-level tracking to two-level tracking as follows: A *level-1 target-to-track hypothesis* is a possible realization of a one-to-one random function from $\Lambda_k^{(1)}$ to I_G defined by

$$\Omega_k^{(1)}(T_k^{(1)}(i)) = i \tag{4.11}$$

and a *level-0 target-to-track hypothesis* is a possible realization of a one-to-one function from $T_k^{(1)}(i)$ to $\{1,...,N_i\}$ (given $T_k^{(1)}(i)$) defined by

$$\Omega_k^{(0)}(T_k^{(0)}(i_1,i_0);T_k^{(1)}(i_1)) = i_0 \quad . \tag{4.12}$$

As in the theory of single-level multitarget tracking, whenever we must distinguish a realization of $\Lambda_k^{(l')}$ from that of $\Omega_k^{(l')}$, we call the former *data-to-data hypothesis.*

4.2.2 General Results

We will derive a recursive formula for calculating each level-1 hypothesis. The results are an extension of the single-level tracking results. For the rest of this section, we make the standard set of assumptions: (1) Targets are interchangeable *a priori.* (2) The data sets are conditionally independent given the target states. (3) The assignment functions are totally random. The first step is a straightforward recursive formula

$$P(\Lambda_k^{(1)} \mid Z^{(k)}) = \frac{P(Z^{(k)},\Lambda_k^{(1)} \mid Z^{(k-1)},\Lambda_{k-1}^{(1)})}{P(Z^{(k)} \mid Z^{(k-1)})} P(\Lambda_{k-1}^{(1)} \mid Z^{(k-1)}) \quad . \tag{4.13}$$

The numerator on the RHS of (4.13) can be expanded in a way similar to that used for the single-level tracking (as described in [1] and [2]) and yields

$$P(\Lambda_k^{(1)} \mid Z^{(k)}) = \frac{P(\Lambda_{k-1}^{(1)} \mid Z^{(k-1)})}{P(Z^{(k)} \mid Z^{(k-1)})} \tag{4.14}$$

$$\sum_{N_G=\#(\Lambda_k^{(1)})}^{\infty} \frac{(N_G-\#(\Lambda_{k-1}^{(1)}))!}{(N_G-\#(\Lambda_k^{(1)}))!} P(N_G \mid \Lambda_{k-1}^{(1)},Z^{(k-1)})$$

$$\sum_{N^G} \prod_{\tau\in\Lambda_k^{(1)}} \frac{(N_{\Omega_k^{(1)}(\tau)}-\#(\overline{\tau}))!}{(N_{\Omega_k^{(1)}(\tau)}-\#(\tau))!} P(N^G \mid N_G,\Omega_{k-1}^{(1)},\Lambda_{k-1}^{(1)},Z^{(k-1)}) \, \mathbf{L}_k(z(k) \mid A_k,N^G,Z^{(k-1)})$$

where

$$N^G = (N_G,N_1,\ldots,N_{N_G}) \tag{4.15}$$

and

$$\mathbf{L}_k(z(k),A_k,N_G \mid Z^{(k-1)}) = \frac{N_{FA}(k)!}{N_M(k)!} \tag{4.16}$$

$$\int P\left((y_j(k))_{j=1}^{N_M(k)} \mid A_k, N_M(k), x(t_k), N^G\right) P_M(N_M(k) \mid I_{DT}(k), x(t_k), N^G)$$

$$P(I_{DT}(k) \mid x(t_k), N^G)\, P(dx(t_k) \mid N^G, \Omega_{k-1}^{(0)}, Z^{(k-1)}, \Omega_{k-1}^{(1)}) \quad .$$

The updating formulae for $P(dx(t_k) \mid N^G, \Omega_k^{(0)}, \Lambda_k^{(1)}, Z^{(k)})$, $P(N^G \mid N_G, \Omega_k^{(1)}, \Lambda_k^{(1)}, Z^{(k)})$ and $P(N_G \mid \Lambda_k^{(1)}, Z^{(k)})$ can be derived in a similar way.

4.2.3 I.I.D.-Poisson Groups

In single-level tracking, an appropriate set of independence assumptions enables us to reduce a general form into a more implementable form. We will repeat such a process for two-level multitarget tracking. The additional assumptions are as follows:

[1] Given the number N_G of groups, the group states tuple $(x_i)_{i=1}^{N_G}$ is a system of independent Markov processes which share common joint probabilities. Thus the state component x_0 common to all the groups is ignored. The number N_G of groups has a Poisson distribution with mean ν_0.

[2] Each $x_i = (x_{i0}, x_{i1}, \ldots, x_{iN_i})$, given N_i, is a stochastic process such that $(x_{ij})_{j=1}^{N_i}$ is a system of interchangeable Markov processes.

[3] The detection is target-wise independent, i.e., the detection of target (i_1, i_0) depends only $(x_{i_10}, x_{i_1i_0})$ and we have

$$P(I_{DT}(k) \mid x(t_k), N^G) = \tag{4.17}$$

$$\prod_{(i_1,i_0)\in I_T} p_D(x_{i_10}, x_{i_1i_0} \mid k)^{\delta(i_1,i_0)} (1 - p_D(x_{i_10}, x_{i_1i_0} \mid k))^{1-\delta(i_1,i_0)}$$

with a common *detection probability function* p_D.

[4] Measurement errors are also target-wise independent, i.e., the value of a measurement originating from a target (i_1, i_0) is correlated only to $(x_{i_10}, x_{i_1i_0})$. The number of false alarms and their values are independent of the targets and from data set to data set. Thus we have

$$P(N_M(k) \mid I_{DT}(k), x(t_k), N^G) = p_{N_{FA}}(N_M(k) - \#(I_{DT}(k)) \mid k) \tag{4.18}$$

and

$$P((y_j(k))_{j=1}^{N_M(k)} \mid A_k, N_M(k), x(t_k), N^G) = \tag{4.19}$$

$$\Big(\prod_{(i_1,i_0)\in Dom(A_k)} p_M(y_{A_k(i_1,i_0)}(k) \mid x_{i_10}(t_k), x_{i_1i_0}(t_k), k)\Big)\Big(\prod_{j\in J_{FA}(k)} p_{FA}(y_j(k) \mid k)\Big)$$

with a *number-of-false-alarm probability function* $p_{N_{FA}}$, a *target-state-to-measurement transition probability density function* p_M, and a *false-alarm-value probability density function* p_{FA}.

Under these assumptions, we can derive results which are very analogous to those in single-level tracking (described in [1] and [2]) and are summarized as follows: (1) Given a level-1 target-to-track hypothesis, the posterior distributions of the group states $(x_i)_{i=1}^{N_G}$ are independent, (2) the posterior distribution of undetected groups becomes Poisson, and (3) the hypothesis evaluation can be reduced to the evaluation of level-1 track-to-measurement likelihood as

$$P(\Lambda_k^{(1)} \mid Z^{(k)}) = \frac{P(_{k-1}^{(1)} \mid Z^{(k-1)})}{P(Z^{(k)} \mid Z^{(k-1)})} \frac{\exp(\nu_k - \nu_{k-1})}{N_M(k)!} \tag{4.20}$$

$$L_{FA} \prod_{\tau\in\Lambda_k^{(1)}} L(y\{\tau \mid k\} \mid Z|_\tau^{(k-1)})$$

where L_{FA} is the false alarm likelihood,

$$y\{\tau \mid k\} = \{y_j(k) \mid (j,k)\in\bigcup\tau\} \tag{4.21}$$

is the set of measurements assigned to level-1 track τ and $L(\cdot \mid Z|_\tau^{(k-1)})$ is the level-1 track-to-measurement likelihood. The forms of the above likelihood functions are very similar to that of the hypothesis evaluation formula for the single-level tracking of dependent targets.

4.2.4 An Example

A straight forward extension of single-level tracking to two-level tracking is possible using the results shown in the previous two subsections. In two-level tracking, however, the combinatorial problem is even more severe, which may make a straightforward extension of single-level trackers infeasible in many applications. For this reason, we may have to develop new techniques for overcoming the additional combinatorial burden inherent to two-level tracking. In this subsection, we will discuss this aspect of the problem in terms of a simple example.

We consider tracking of groups of ground vehicles moving on a road network. By a two-step transformations to take care of (1) the route selection by each group and (2) the curvature of each road segment, the problem can be reduced to that of tracking groups of targets moving on a straight line. Let u_i be the 1-dimensional position of the lead vehicle of the i-th group and v_i be its velocity. Then the position and the velocity of the j-th vehicle in group i can be modeled as

$$u_{ij} = u_i - (j-1)c_i v_i + \xi_{ij} \tag{4.22}$$

and

$$v_{ij} = v_i + \eta_{ij} \tag{4.23}$$

where $c_i v_i$ is the expected distance between two vehicles in group i, ξ_{ij} and η_{ij} represent randomness in position and velocity of each vehicle in the group. We assume that the randomness can be modeled by independent gaussian random variables. The group dynamics are then assumed to be a simple almost constant velocity model with an appropriate white gaussian driving noise. Thus we may have a very simple target model in which the state component common to all the targets in group i is

$$x_{i0} = (u_i, v_i, a_i) \tag{4.24}$$

where a_i is a discrete variable representing the *type* of group i. The individual target state of the j-th target in group i is then simply its *type* a_{ij}.

For each possible type of group, we assume that we have a sufficient number of *templates* of the group including composition of different types of vehicles and their order when moving on the road. Each template can be represented by

$$e = (a, N, b_1, \ldots, b_N) \tag{4.25}$$

where a is the type of a group, N is the number of vehicles in the group and b_i is the type of the i-th vehicle in the group. Therefore the level-1 track distribution, i.e., the group state distribution, is a distribution on $(-\infty,\infty)^2 \times E$, where E

is the set of all the templates. In general, we may assume at least in an approximated sense the independence of motion from the type component as $P(du, dv, de) = P(du, dv)P(de)$.

When a data set is received from a sensor, each group hypothesis is given a set of measurements which may be associated to it. Then the set of measurements is ordered linearly and, for each template, the level-1 track-to-measurement likelihood is calculated after template-to-measurement matching as shown in Figure 4-6. In such a process, we must use a very effective method for determining a likely level-1 track-to-measurement assignment. For example, for each template, we first estimate the most probable distance between targets based on the velocity estimate and then spread the vehicles in the template accordingly. Then, by an effective assignment algorithm, we can find a feasible assignment between the given set of measurements and the vehicles in the template. After determining the assingment, we can calculate the level-1 track-to-measurement likelihood.

4.2.5 Distributed Hypothesis Formation and Evaluation

As shown earlier in this report, distributed hypothesis formation is a process of creating a logically consistent set of hypotheses from a collection of local sets of hypotheses. This process amounts to the consistency checking on the overlapped pieces of information in the past. It is also determined purely by the definitions of tracks and hypotheses and independent of their probabilistic nature. Therefore it is expected that we may extend the single-level tracking results to the two-level or in general level-l tracking cases. The results may be a similar type of consistency checking on the predecessors of tracks and hypotheses. However, although the final results are fairly simple in single-level tracking cases, complicated steps were necessary to derive implementable results. It is hence expected that the logical arguments involved in two-level tracks and hypotheses may well be very complicated.

On the other hand, distributed hypothesis evaluation involves the distributed estimation and is highly dependent on the structure of the global hypothesis evaluation formula. In the single-level tracking cases (with the i.i.d. Poisson assumption), the hypothesis evaluation equation is, in essence, a product of track likelihoods and each track likelihood is an integration of a product of state-to-measurement transition probability densities. Thus each track likelihood can be decomposed using distributed estimation theory. In two-level tracking

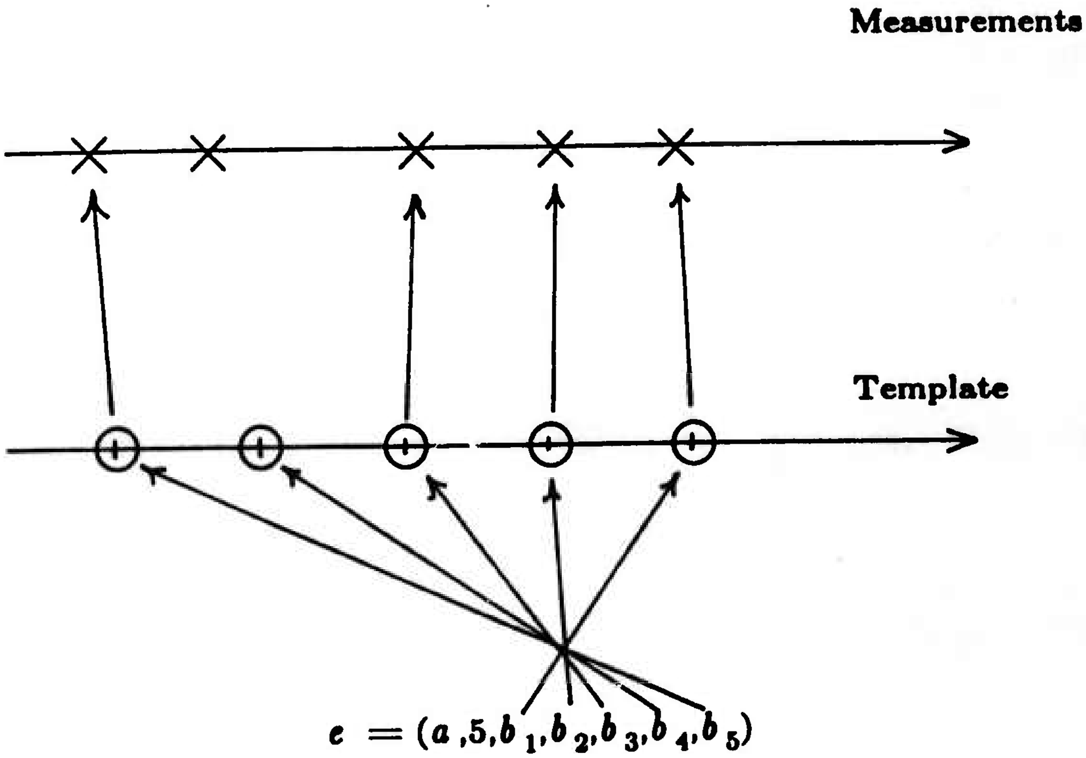


Figure 4-6: Template-to-Measurement Matching

cases, however, the level-1 track-to-measurement likelihood involves summation over many possible numbers of targets in each group, which may cause difficulty in decomposing the track likelihood into the independent components. We may well need a kind of aggregation of tracks and hypotheses in order to produce a workable algorithm for distributed hypothesis evaluation for the two-level tracking. The dynamic behavior of groups may also complicate the discussions.

4.3 CONCLUSION

A first-cut analysis on multitarget tracking concerning a structured set of targets has been discussed in this section. The discussions in this section are summarized as follows: (1) Structured sets of targets may be treated in an integrated form and concepts of tracks and hypotheses can be extended from the single-level cases in a straightforward way. (2) Two-level multitarget tracking hypothesis evaluation can be done by extending the single-level tracking results. (3) Practical methods for implementing two-level hypothesis evaluation needs however further investigation. (4) Distributed hypothesis formation and evaluation for two-level tracks and hypotheses may be possible by extending the single-level results but we need more time to clear this problem. The future efforts pertaining to the topics covered in this section may include: (1) effective implementation of single-level tracking with correlation among targets, (2) implementation of two-level multitarget tracking algorithms, and (3) development of distributed level-1 hypothesis formation/evaluation algorithms.

5. ACOUSTIC TRACKING EXPERIMENTS AND ALGORITHMS

As part of the DARPA DSN program, M. I. T. Lincoln Lab. has performed research on the tracking of low flying aircraft using acoustic sensors. A DSN test bed has been developed and used to test and demonstrate DSN techniques and technology. This section describes how the general distributed tracking algorithms developed at ADS can be applied to the acoustic tracking scenarios used by Lincoln Lab. We first present some candidate experiment scenarios. Then we discuss the development of acoustic tracking algorithms based on the multiple hypothesis approach.

5.1 ACOUSTIC TRACKING EXPERIMENTS

This section describes the kind of DSN systems and scenarios for which new distributed acoustic tracking algorithms are to be developed and evaluated. It describes a family of systems and scenarios that will result in a range of tracking problems; from easy to quite difficult. The system and scenarios are essentially those which Lincoln had considered in a more informal way during its earlier algorithm development effort and which have been used as the basis for the evaluation of existing Lincoln algorithms. Much of the information in this section has been provided by Lincoln Lab.

The system follows that of the Lincoln test bed and is a multiple node acoustic system for low flying aircraft surveillance. The sensors at each node are small acoustic arrays that provide lists of possible target detections along with azimuth, accuracy and power level estimates every few seconds. The scenarios range from single aircraft with straight flight paths operating under low background noise conditions to more difficult scenarios involving several maneuvering aircraft. Basic communication service consists of an unacknowledged radio broadcast service in which nodes can receive broadcasts only from a limited set of neighbors.

5.1.1 Acoustic Sensors

The acoustic sensors in the DSN test bed are small microphone arrays which detect possible targets, measure acoustic azimuths and provide signal-to-noise estimates that can be used to ascribe accuracy values to the azimuth measurements. Measurements are nominally made every two seconds and the measurement (after signal processing) corresponds to an average target azimuth over a two second interval. The sensors provide no target elevation information. The development of tracking algorithms will emphasize acoustic arrays with capabilities comparable to those that Lincoln had been using, with at most a limited consideration of arrays with different performance characteristics.

The azimuth accuracy of the acoustic arrays is on the order of two degrees and, depending upon target type and background noise conditions, the detection range for a single target is from a few to a few tens of kilometers. A good nominal value to use is five kilometers. Target detection probability and azimuth accuracy depend upon signal to noise ratio. For a given signal source strength the signal-to-noise ratio depends upon range although it is also influenced by topography and propagation conditions. Detection probability will be low at long range and increase as the target comes closer to the sensor. Topographic features such as hills may introduce quiet zones within which this increase of signal level with decreasing range does not hold.

The number of targets within the detection range of a sensor that can be simultaneously detected and isolated in azimuth depends upon many factors. These include array aperture, number of sensors in each array, noise level, signal level and the azimuthal separation of the targets. For existing DSN arrays and signal processing algorithms the number of targets that can be simultaneously isolated is on the order of three to five, assuming they are sufficiently separated in azimuth and do not have excessively disparate power levels. A nominal value for the required azimuthal separation for targets with roughly equal power is 20 degrees.

The number of targets that can be isolated by a single acoustic sensor limits the local target density for target tracking but does not limit the total number of targets for the entire system. For a fixed spatial density of DSN nodes the number of targets that can be individually tracked will scale up linearly with the geographic area of the network. In addition, clusters of unresolved targets can be tracked if not isolated from each other.

In the absence of targets, the number of false detections generated by the sensor and its associated signal processing algorithms is on the order of three to five for each measurement interval.

Measurement intervals other than two seconds might be used to improve azimuth measurement accuracy for fast-moving nearby targets or to improve accuracy and detectability for very distant targets. If this is a controllable variable, it may used for tracking maneuvering targets. Also, the sampling rate may be decreased when the tracking performance is adequate or when the algorithm cannot keep up with the arrival of the measurements.

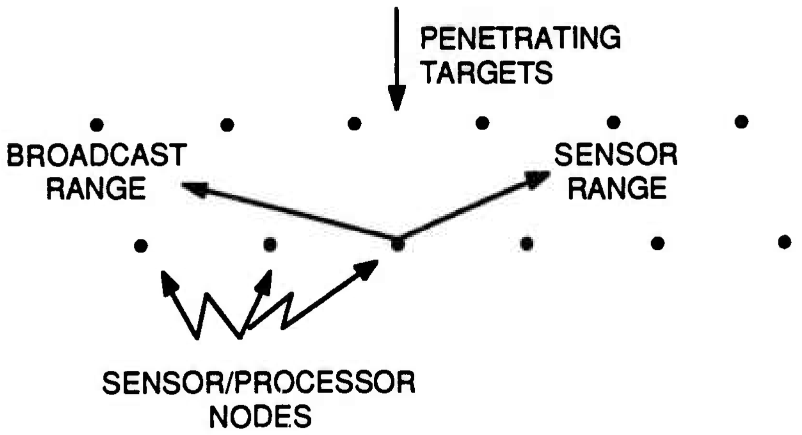
A lower limit to azimuth accuracy is imposed by propagation physics, not by acoustic array size or processing techniques. A reasonable value for this lower limit is probably about one degree. Azimuth errors larger than a few degrees might be obtained from arrays that are smaller than the five meter arrays used by Lincoln or for signals with only very low frequency signal content. Errors of more than about ten degrees probably should not be considered unless very poor location accuracy is acceptable. In general, we will be concerned with systems which can locate aircraft to within a kilometer or less in the horizontal plane.

Specific and detailed statistical sensor models will be formulated and refined as needed to support the development of tracking algorithms.

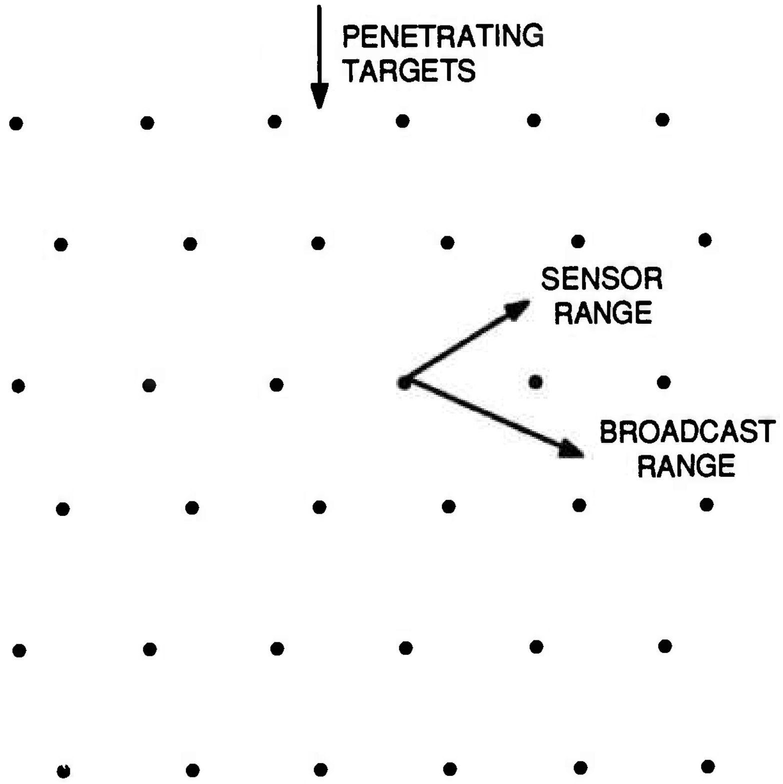
5.1.2 System Deployment

There are two primary DSN deployment options: *barrier* and *area*. The barrier concept is to deploy a thin DSN over a long linear extent and is appropriate for early warning situations. In this case, the primary interest is in target acquisition and transient phenomena as the targets approach and pass through the barrier. Area deployment is needed to provide continuous surveillance over large areas and steady-state performance for targets internal to the network is a more important issue. Both *transient* and *steady-state* performance issues come together in the outer layers of an area DSN or in a barrier system with more than one layer of thickness. Transient behavior will dominate as targets approach the outer layer of nodes and steady-state behavior should dominate by the time the second or third layer is reached. The DSN systems of interest contain many nodes but it should be possible to investigate and test algorithms with from two to six node configurations.

Examples of barrier and area DSN systems are shown in Figure 5-1. The deployment shown in the figure is on a regular grid but in general deployments will be more randomized and algorithms should be designed with that in mind. Depending upon the specific system and deployment, the number of nearest neighbors at about the same distance will range from two to six.



(a) BARRIER SYSTEM



(b) AREA SYSTEM

Figure 5-1: DSN Deployment Configuration

Sensor detection range, distances between nodes, and the broadcast communication range are very important DSN network parameters. Lincoln tracking algorithms have been developed for the case when the sensor range and the distance between sensors are about equal and the nominal broadcast range is twice the distance between sensors. This situation appears to be optimal. For smaller broadcast ranges, the nodes will become information poor unless additional communication mechanisms are added to distribute information to more remote nodes. For smaller sensor ranges, the nodes become information poor but the problem is more fundamental since that lack of information is because not enough sensors detect each target at the same time. The new algorithms should also emphasize the nominal case but may offer performance improvement possibilities under one or both of the information poor situations.

5.1.3 Internodal Communication

The nominal internodal communication is a limited range unacknowledged broadcast. Ideally a broadcast is received without error by all nodes within range. The nominal broadcast area is a circle defined by the broadcast range.

Algorithms must also operate under other than idealized situations. There may be "dead areas" within the nominal broadcast disks. Those dead areas may be known and accounted for by tracking algorithms. They may be unknown and the algorithms should be designed to be robust in their presence if possible.

Communications may also be subject to errors in the form of lost messages. Algorithm development should consider how to treat randomly lost broadcast messages. Note that the messages will be lost upon reception, not broadcast, so that a message may be correctly received by any number of nodes within broadcast range. The system should be designed to handle at least a few percent of lost messages. In general the performance may degrade with increasing percentages of lost messages, but the system should not catastrophically fail.

5.1.4 Target Scenarios

The number of possible target scenarios is very large. The scena[illegible], to be used in the experiments will be selected based upon Lincoln's past experience with trackers and our best judgement concerning interesting or critical situations. To be consistent with the Lincoln scenarios, we will assume the targets to be low flying at about 500 meters above ground level.

The scenarios are described in terms of the local density of targets. One, two, and three target situations are emphasized. Satisfactory local performance for resolved and unresolved targets will translate directly into satisfactory performance for many more targets and clusters of targets in a large DSN system.

Initially we will consider single target scenarios. This is primarily for testing the basic algorithms and the communication between nodes. Both maneuvering and non-maneuvering situations should be considered. Maneuvers include changes of speed and direction. For non-maneuvering targets, we should consider both direct and angled approaches to the DSN boundary and with targets passing very near to nodes as well as between nodes. At least some consideration should be given to high speed (Mach 0.9) cases to be certain that there is no unexpected behavior. The case of Mach 0.6 is probably more typical of low flying aircraft or long range cruise missiles. Mach 0.1 is representative of a very slow target such as a helicopter.

Since our algorithms are supposed to handle multiple targets effectively, we will consider two-target scenarios of varying complexity. We will use a set of two-target scenarios covering three distinct target configurations. These are the in-line formation in which the two targets follow one behind the other, the parallel formation in which they follow parallel tracks and the crossing formation in which their tracks cross. In most cases, the targets will have the same speeds but some situations with targets at two different speeds are included since a fast target overtaking a slow one may cause some problems for tracking algorithms. The crossing scenario will be most stressing when targets reach the crossing point (at different altitudes) at the same time.

One and two target scenarios may not adequately stress track initiation and data association algorithms although they should be adequate for most other purposes. Thus we will experiment with a few additional three-target scenarios that will provide additional stress.

5.1.5 Measures for Performance Evaluation

In this section, we discuss suitable measures for evaluating the performance of distributed acoustic tracking algorithms. The evaluation is complicated by the presence of multiple nodes, multiple targets, and multiple hypotheses since at any particular time, the tracking performance depends on the node, the hypothesis, and the target of interest. In addition to local measures, more aggregate measures for the entire system are also desirable. The system performance also depends on the sensor and target characteristics in addition to the system configuration. Contributions to the performance from these parameters should be isolated for a fair evaluation of the algorithm performance.

5.1.5.1 Evaluation of Tracking performance

We first discuss measures which evaluate how well the algorithms track. Since a multiple hypothesis approach is used, the tracking performance can be divided into two levels: hypothesis level and track level. These will be discussed separately in the following:

1. *Hypothesis-level* measures include the number of false targets and the number of missed targets. A hypothesis is a collection of tracks. Using appropriate thresholds, the tracks can be identified with the actual targets in the scenario. The *missed targets* and the *false tracks* can thus be enumerated. This can be performed for the best hypothesis (one with the highest probability) or it can be evaluated for all hypotheses to obtain an expected number of missed targets and false tracks.

2. *Track-level* measures include the *estimation error* (e.g., RMS error) for the detected targets. The presence of multiple targets implies that some average over the targets has to be considered unless individual target errors are to be represented explicitly. For multiple hypotheses, the error for each target may be that of the best hypothesis or it can be evaluated over all hypotheses to obtain an expected error.

Both of the measures discussed above may be evaluated for a given time for each node. In fact, it is frequently desirable to consider these measures as a function of time for each node to see how each node performs with time. A node's performance may fluctuate with the quality and quantity of data available to the node at a particular time. The average performance of the node can be calculated by averaging these measures over time.

The performance of the overall system depends on that of all the nodes in the system. One may want to assume a fusion node which collects information from all the DSN nodes and use it to measure the performance of the overall system. As discussed before, the sensor and target characteristics are also important variables affecting the system performance. If poor sensors are used with a difficult target scenario, then the performance of the system would be expected to be poor inspite of a good algorithm. Thus a reference should be used in evaluating algorithm performance. A suitable reference is the performance of the centralized algorithm and that of distributed algorithm can be compared with it.

5.1.5.2 Cost measures

The cost measures reflect the amount of resources used to produce the measured performance. Relevant resources include the local computation time, the memory size and the amount of communication. The effectiveness of a processing node or the total system can be measured by the amount of resource it needs to produce a fixed level of performance. When there is a constraint on the resource, a good system is one which uses the resources to produce the best performance.

In a simulation of the DSN, ways should be devised to measure the resource utilization, e.g., the computation time, the memory size profile, the amount of communication, etc. The computation time required may be recorded by means of a system clock. The memory size can be expressed in bytes or in terms of the number of hypotheses or tracks stored at each node. The amount of communication can also be measured in various ways. These quantities can be expressed for each node at each time, or for each node over an interval. From these system level measures can be computed.

Such data may provide vital information on possible refinements or improvements of the various modules. If necessary, resource allocation modules may be developed to meet the hard constraints at each DSN node. For example, a way for the processing to keep up with the arrival of data is by skipping sensor scans when the processing lacks behind arrival of data.

All the measures described above can be evaluated using simulated or real data. Monte Carlo simulations can be performed when the data are synthetic. Statistics on the various measures can then be obtained.

5.1.6 Data Sources

Real and simulated acoustic data will be used to test algorithm components as they are developed.

Real data will provide stressing clutter situations for simple scenarios with only a few nodes and with poor to good knowledge of ground truth. Data recorded at Lincoln Laboratory and at test ranges and air bases in the Western U.S. are available.

Many scenario variations will require use of simulated data. That data will be generated using a previously developed acoustic data simulator that operates under VAX/UNIX developed by Lincoln Lab. Given system parameters and a description of the target scenario, it simulates the clutter and azimuth measurement outputs from the nodal signal processing subsystems. Using this tool we will, under very controlled conditions, generate algorithm development data for situations that would otherwise be very difficult to obtain. We plan to implement this data generator on the same machine as the algorithms to facilitate experimentation.

5.2 ACOUSTIC TRACKING ALGORITHMS

Although the general methodology developed in our research is in theory applicable to the acoustic tracking problem, the acoustic scenario raises technical issues which need to be addressed before algorithms can be developed to perform satisfactorily. In the following sections we discuss these issues and relevant models and how the general algorithms can be adapted for acoustic tracking.

5.2.1 Issues and Models in Acoustic Tracking

Some special features of acoustic tracking and the associated technical issues are:

- *Azimuth-only measurements.* Each acoustic sensor measures only the azimuth of the target. Thus from a single node, the target location is not very observable from the azimuth measurements. From a pair of nodes, however, a target becomes more observable. An important question is thus the types of processing to be performed locally by one node and jointly by a pair of nodes. One possibility is to use different representations such as azimuth tracks for local processing and position tracks after fusion.

- *Propagation delay.* Acoustic signals generated by a target do not reach a node instantaneously. Since the target speed is substantial compared to the speed of sound in the air, the delay has to be considered explicitly in any information processing. For example, the true bearing of a target at a node can be quite different from its apparent bearing at the node.

- *Poor sensor resolution.* Due to the poor sensor resolution (20 degrees separation needed before two targets can be distinguished), two targets which are close together may be detected as a single target. Our previous discussion has largely ignored this possibility. New techniques will have to be developed to handle this situation.

- *Range dependent detection.* Since target detection depends on the range, and range affects the sound pressure received at a node, some useful information may be present in the sound pressure. On the other hand, the acoustic propagation characteristics in air may be too complicated and unreliable. The question is whether this intensity information can be exploited or not, and if yes, how it can be exploited.

Based on the scenarios described in Section 4.1, we have assumed the following target and sensor models. Since the targets are assumed to be flying low, their altitudes are ignored and they are modeled as objects moving in the 2-dimensional space. The motion is modeled by constant velocity or constant acceleration (and if necessary constant jerk). The target maneuvering is modeled by additional white noise excitation to the target dynamics.

The sensor model follows that of [13] which documents the synthetic data generator developed and used for simulation by MIT Lincoln Laboratory. Let a target position viewed from a sensor (located at the origin) at time t be $x(t)$. The sound wave received at time t by the sensor originated from the target at time $t-\delta$, where the time delay δ is determined by

$$\|x(t-\delta)\| = c\,\delta \tag{5.1}$$

where

- $\|\cdot\|$ is the Euclidean norm of a vector

- c is the speed of the sound in the air (See Fig. 5-2).

Equation (5.1) has a unique solution δ provided $x(\cdot)$ is differentiable and $\|\dot{x}(t)\| < c$ (subsonic). It determines the acoustic azimuth (measured clockwise from the north) ϕ of the target with respect to the sensor. The measured acoustic azimuth ϕ_M contains measurement error as

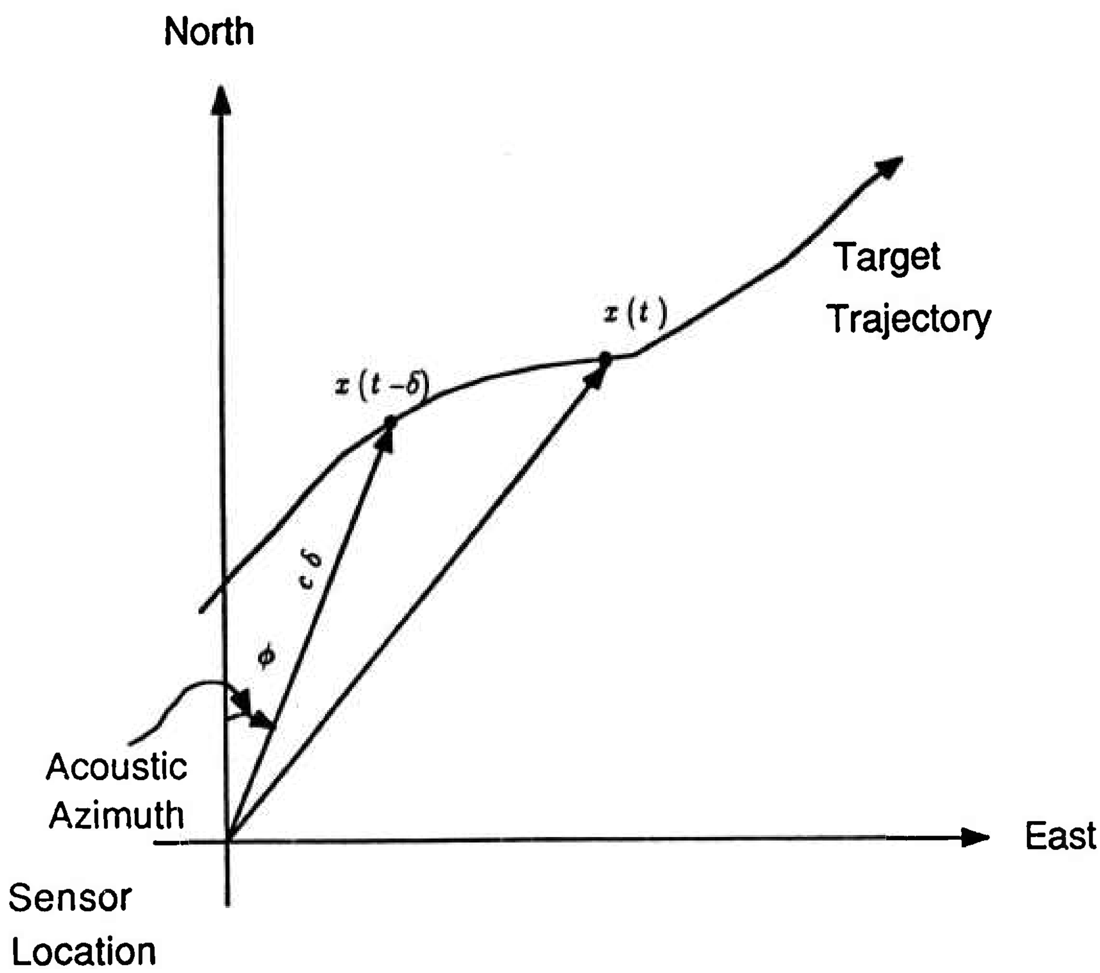


Figure 5-2: Target Sensor Geometry

$$\phi_M = \phi + w \tag{5.2}$$

where w is modeled by an independent zero-mean gaussian random variable (r.v.) whose variance is yet to be specified.

Let the sound pressure at the 1-meter distance from the target be s_0. Then the sound pressure measurement s_M at the sensor is

$$s_M = G \frac{s_0}{r^2} \tag{5.3}$$

where r is the acoustic range, i.e., $r = c\,\delta$, and G is the sensor gain. To account for irregular propagation and other random factors, either additive or multiplicative noise should be added to (5.3). The sensor also measures ambient noise s_N. Thus when the measured sound pressure exceeds a given threshold s_{TH}, the sensor generates a measurement $y = (\phi_M, s_M, s_N)$ consisting of the acoustic azimuth and the signal/noise sound pressure levels.

The azimuth measurement error standard deviation (SD) σ_ϕ (of w in (5.2)) is determined by

$$\sigma_\phi = \frac{\delta\phi}{\Gamma(SNR)} \tag{5.4}$$

where

- $\delta\phi$ is the sensor resolution (about 20 degrees)

- $SNR = s_M / s_N$

- $\Gamma(SNR) = \min\{\max\{1, \sqrt{SNR}\}, 10\}$.

The number of false alarms is modeled as a Poisson random variable independent from scan to scan. The delayed azimuth value of a false alarm is distributed uniformly on $[0, 2\pi]$ and the sound pressure value has an exponential distribution biased by the threshold value.

When two acoustic azimuth measurements, ϕ_M^1 and ϕ_M^2, are close enough, i.e., $|\phi_M^1 - \phi_M^2| < \delta\phi$, they are merged into a single measurement. The merged acoustic azimuth measurement is modeled as

$$\phi_M^m = q\,\phi_M^1 + (1-q)\phi_M^2 \tag{5.5}$$

where

$$q = \begin{cases} 1 & \text{if} \quad s_M^1 > 5 s_M^2 \\ \dfrac{s_M^1}{s_M^1 + s_M^2} & \text{otherwise} \end{cases}, \tag{5.6}$$

and s_M^i is the unmerged sound pressure measurement corresponding to ϕ_M^i. The merged sound pressure measurement becomes

$$s_M^m = \begin{cases} s_M^1 & \text{if} \quad s_M^1 > 5 s_M^2 \\ s_M^1 + \dfrac{1}{2} s_M^2 & \text{otherwise} \end{cases} \tag{5.7}$$

In Equations (5.5) to (5.7), we have assumed that $s_M^1 > s_M^2$. Otherwise we should exchange the indices 1 and 2. Note the nonlinear nature of the merged measurement model. When one measurement is much stronger than the other one, the merged measurement is dominated by the stronger measurement.

5.2.2 Nodal Structure and Tracks

The general nodal architecture of Section 2 applies to acoustic tracking without much modification. Figure 5-3 shows a more detailed functional architecture of each node and results from integrating Figures 1-2 to 1-4 in Section 1. Each node contains a local data base of hypotheses which is updated whenever new information arrives. This can happen in either one of two ways: data arriving from the local sensors or messages arriving from the other nodes. The two corresponding updating functions are then local information processing and information fusion.

The hypotheses in the tracking data base are the same as defined in Section 2. Each hypothesis consists of a set of tracks and represents a feasible explanation of the origins of the measurements. Each track τ is accompanied by a *target state distribution (TSD)* which represents the distribution $p(x_t \mid \tau, Z)$ of target state x_t conditioned by the track τ and the cumulative sensor information Z. Because of the nature of acoustic sensors, these track state descriptions depend on the number of sensors involved. For tracks formed from the measurements of a single sensor, the TSD is not very informative since the target state is not very observable from one acoustic sensor. On the other hand, more information about the target can be extracted from the measurements of multiple sensors.

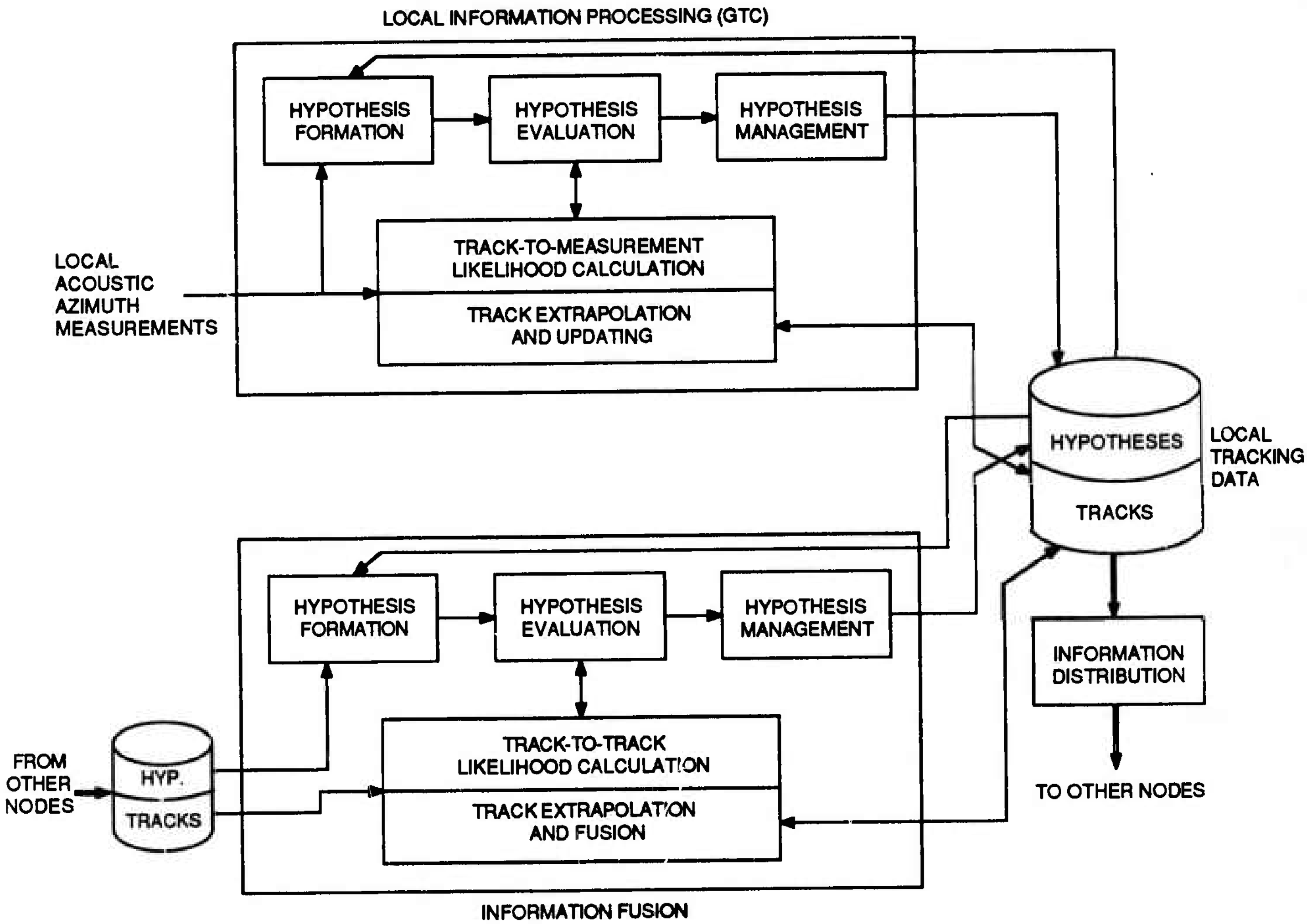


Figure 5-3: Functional Architecture of Each Node

Each target state distribution (TSD) consists of *geolocational TSD (GTSD) component(s)* and a *sound pressure TSD (SPTSD) component.* A TSD component is called *local* if the distribution can be derived from the measurements of a single sensor; otherwise it is called *global.* Thus a GTSD component is either global or local while a SPTSD component is always local. In general GTSD or SPTSD components may be represented as sum-of-gaussians (i.e., multiple gaussian terms with a probabilistic weight attached to each term). A track may have only a local GTSD component. In such a case, a track is said to be *local.* Or a track may have a global GTSD component or both global and local GTSD components. Then the track is said to be *global.* Figure 5-4 shows a taxonomy of the target state distributions.

A local GTSD component is a gaussian distribution on the (local) acoustic azimuth of a target and its derivative, $(\phi,\dot{\phi})$, and possibly higher-order derivative(s). Local GTSD components are used to allow each sensor to initiate tracks locally from acoustic azimuth measurements. As a local track accumulates acoustic azimuth data, the acoustic azimuth rate $\dot{\phi}$ is estimated with increasing accuracy as indicated by the decreasing variance matrix in the local GTSD component terms. A global GTSD component term is a gaussian distribution on the

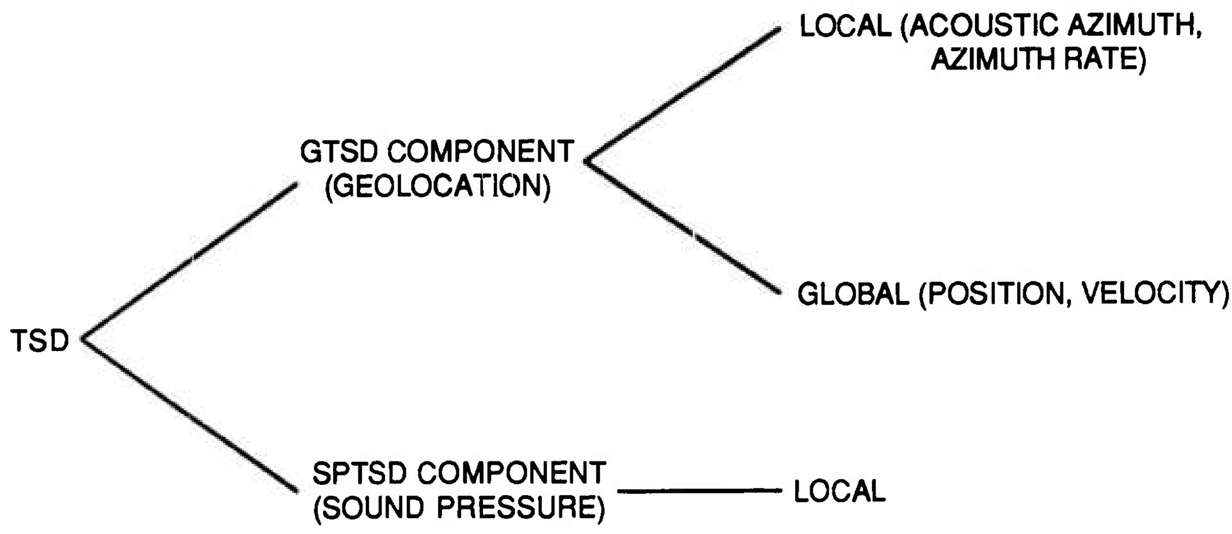


Figure 5-4: Target State Distributions

global coordinates, i.e., the target position and the velocity in the north-east coordinate, and possibly their higher-order derivatives. A global GTSD component term is formed from two local tracks when different sensors communicate.

A SPTSD component tracks the change in the measured sound pressure. It serves the following functions:

- to provide an additional discriminant (particularly from false alarms)
- to predict when a target leaves the sensor coverage
- to predict the merged acoustic azimuth measurement when measurement merging is likely.

This component is also used to estimate the targets' noisiness. A SPTSD component term is a gaussian distribution on the (fictitiously noiseless) received sound pressure s and its derivative, and possibly higher-order derivative(s). The actually measured sound pressure s_M is modeled by

$$s_M = s + w_s \tag{5.8}$$

where the artificial noise term w_s (modeled by independent zero-mean gaussian r.v.) accounts for scan-to-scan fluctuation of the sound pressure measurements. It may be argued that a multiplicative noise term is more appropriate. In such a case, (5.8) remains valid after taking the logarithm of each variable.

The updating of each TSD component is performed simultaneously with the hypothesis evaluation (described in the next section). On the other hand, the extrapolation of each TSD component term is performed with the help of appropriate target dynamic models, i.e., constant-velocity or constant-acceleration linear models with an appropriate white noise input. For example, in order to update a local GTSD component term, we may use the following simple set of differential equations:

$$\frac{d}{dt}\phi = \dot{\phi}$$

$$\frac{d}{dt}\dot{\phi} = white \;\; noise \tag{5.9}$$

Since the acoustic azimuth dynamics are in fact nonlinear, the intensity of white noise must be chosen to compensate such nonlinearity in addition to any target maneuvering if necessary.

5.2.3 Local Processing

As discussed in Section 2, as new measurements arrive from the sensors, the local tracking data base is updated by hypothesis processing. In the following we discuss how hypothesis formation and evaluation are adapted for acoustic sensors.

5.2.3.1 Hypothesis Formation

Our assumption in Section 2 was that no two tracks can share the same measurement. Due to the poor resolution of acoustic sensors, this assumption is no longer valid. In general, two or more targets may give rise to only one merged measurement when they are close to each other (within 10 to 20 degrees). To simplify the discussion here, we assume only two-way measurement merging as modeled in Section 5.2.1.

Consider the arrival of a sensor report. The hypothesis $\overline{\lambda}$ at the node is then to be expanded with the measurements in the new sensor report to form new hypotheses. Before making use of the measurements in the new data set, we must consider the possibilities of some tracks in the hypothesis $\overline{\lambda}$ being merged. Thus the hypothesis is first expanded into the set of *track merging hypotheses*, each representing one possible merging of tracks. Mathematically, track merging hypothesis is a partition $\overline{\Lambda}_m$ of $\overline{\lambda}$ such that $\#(\overline{\mathbf{T}}) \leq 2$ for any $\overline{\mathbf{T}} \in \overline{\Lambda}_m$ where $\#(A)$ is the number of members in a set A.

Each track merging hypothesis is then expanded by the set of measurements as in the usual case when there is no measurement merging. Fig. 5-5 illustrates this two-step hypothesis expansion: first by track merging and next by the measurements. In the figure, a hypothesis $\overline{\lambda}$ having three tracks is expanded into four track merging hypotheses, $\overline{\Lambda}_m^1$ to $\overline{\Lambda}_m^4$, each of which is further expanded by the measurements (shown by shaded triangles in Fig. 5-5). Fig. 5-6 shows the expansion of the hypothesis $\overline{\Lambda}_m^1$ by the two measurements in the current sensor scan.

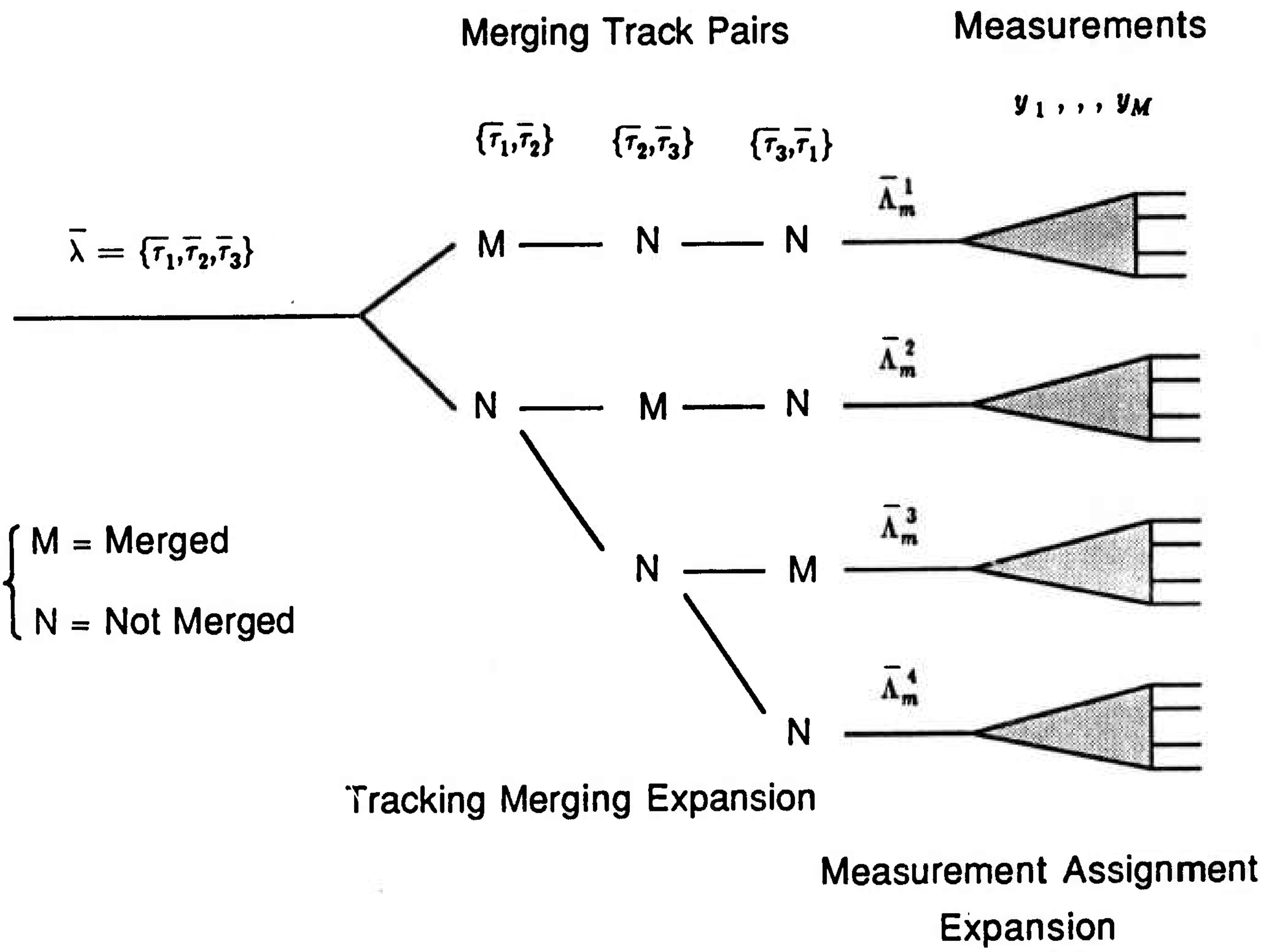


Figure 5-5: Hypothesis Expansion with Measurement Merging Possibility (1)

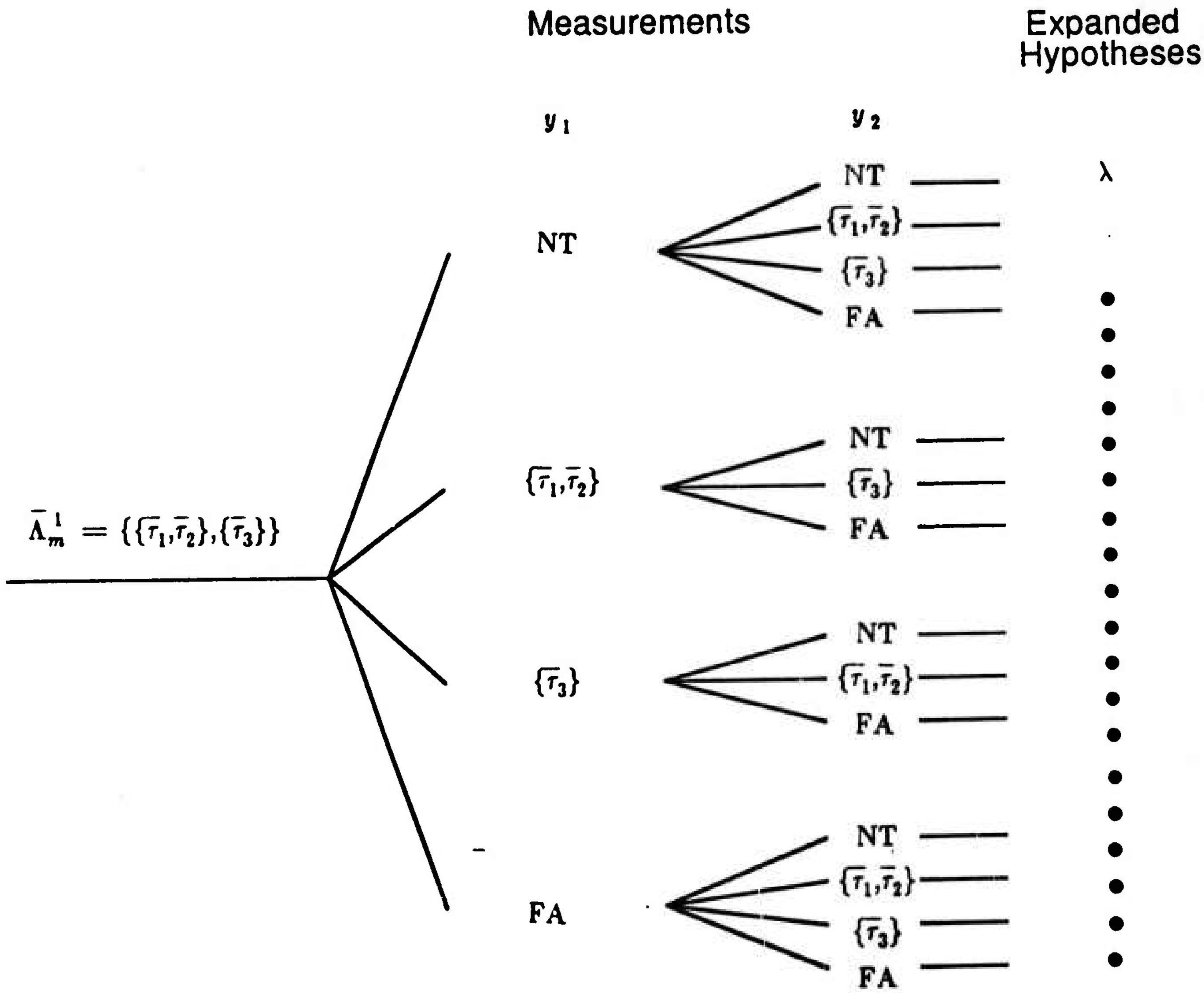


Figure 5-6: Hypothesis Expansion with Measurement Merging Possibility (2)

5.2.3.2 Hypothesis Evaluation

After expanding all the hypotheses $\bar{\lambda}$ in the old cluster the resultant collection of new hypotheses forms an updated cluster. Each new hypothesis λ has a unique parent $\bar{\lambda}$ and a unique track merging hypothesis $\bar{\Lambda}_m$, from which λ is generated. Then evaluation of hypotheses, with the possibility of measurements merging, can be done by replacing $\bar{\lambda}$ by $\bar{\Lambda}_m$ in the general hypothesis evaluation formula [4], and probabilistically assessing the joint event of "two tracks merged and generating a single measurement." The results may be summarized as

$$Prob.\,\{\lambda \mid Z\} = C^{-1}\,Prob.\,\{\bar{\lambda} \mid \bar{Z}\}$$

$$\left(\prod\{L\,(y \mid \bar{\mathbf{T}}) \mid \bar{\mathbf{T}} \in \bar{\Lambda}_m \cup \{\emptyset\} \text{ and } \bar{\mathbf{T}} \text{ is assigned measurement } y\}\right)$$

$$\left(\prod\{L\,(\theta \mid \{\bar{\tau}_1, \bar{\tau}_2\}) \mid \tau_1 \in \lambda \text{ and } \tau_2 \in \lambda \text{ but they did not merge}\}\right)$$

$$\left(\prod\{L\,(\theta \mid \bar{\tau}) \mid \tau \in \lambda \text{ but not assigned any measurement, i.e., } \bar{\tau} = \tau\}\right) \quad (5.10)$$

where

- Z is the cumulative data set including the current sensor scan
- $\bar{Z}$ is Z minus the current sensor scan
- C is the normalizing constant
- θ is the symbol used to represent "no measurement".

The $L\,(\cdot \mid \cdot)$'s are likelihood functions defined below.

Newly Detected Target Likelihood ($L\,(y \mid \emptyset)$). When $\bar{\mathbf{T}}$ is $\emptyset$, $L\,(y \mid \emptyset)$ is the likelihood of measurement y originating from a target undetected before and is given by

$$L\,(y \mid \emptyset) = \beta_{NT}\,(\phi_M)/\beta_{FA} \quad (5.11)$$

where $\beta_{NT}\,(\cdot)$ is the expected density of undetected targets, translated into the acoustic azimuth space, i.e, $[0,2\pi]$. β_{FA} is the density of the false alarms over the $[0,2\pi]$ interval, i.e.,

$$\beta_{FA} = \nu_{FA} / 2\pi \tag{5.12}$$

where ν_{FA} is the expected number of false alarms (about from 1 to 3) per scan. Equation (5.11) also assumes that the sound pressure measurement distribution of a target "heard" (detected) for the first time is equal to that of a false alarm.

Old Track Measurement Likelihood ($L(y \mid \overline{\mathbf{T}})$, $\overline{\mathbf{T}} \neq \emptyset$) When $\overline{\mathbf{T}} \neq \emptyset$, $L(y \mid \overline{\mathbf{T}})$ is the likelihood of measurement y originating from an existing track $\overline{\mathbf{T}} = \{\overline{\tau}\}$ or jointly from two existing tracks $\overline{\mathbf{T}} = \{\overline{\tau}_1, \overline{\tau}_2\}$ and is defined by

$$L(y \mid \overline{\mathbf{T}}) = \frac{L_\phi(\phi_M \mid \overline{\mathbf{T}})}{\beta_{FA}} \frac{L_s(s_M \mid \overline{\mathbf{T}})}{p_s^{FA}(s_M)} \tag{5.13}$$

where

- $L_\phi(\phi_M \mid \overline{\mathbf{T}})$ is the azimuth measurement likelihood

- $L_s(s_M \mid \overline{\mathbf{T}})$ is the sound pressure likelihood

- p_s^{FA} is the probability density of the false alarm sound pressure.

For simplicity, we assume that the GTSD and SPTSD components of each track are both single-termed. There are two different cases.

Case 1: No Merged Measurement. In case $\#(\overline{\mathbf{T}}) = 1$, i.e., when there is no merging, we have

$$L_\phi(\phi_M \mid \{\overline{\tau}\}) = \frac{1}{\sqrt{2\pi}\tilde{\sigma}_\phi} \exp\left(-\frac{1}{2}\left(\frac{\phi_M - \overline{\phi}}{\tilde{\sigma}_\phi}\right)^2\right) \tag{5.14}$$

with $\overline{\phi}$ being the acoustic azimuth prediction by a local or global GTSD component of $\overline{\tau}$ and $\tilde{\sigma}_\phi^2$ being the corresponding innovations variance given by

$$\tilde{\sigma}_\phi^2 = \sigma_\phi(SNR)^2 + \overline{\sigma}_\phi^2 \tag{5.15}$$

where $\sigma_\phi(\cdot)$ is the azimuth measurement error standard deviation as a known function of signal-to-noise ratio $SNR = s_M / n_M$, and $\overline{\sigma}_\phi^2$ is the expected error variance of the azimuth estimate by track $\overline{\tau}$.

The sound pressure likelihood is calculated as

$$L_s(s_M \mid \{\overline{\tau}\}) = \frac{1}{\sqrt{2\pi}\tilde{\sigma}_s}\exp\left(-\frac{1}{2}\left(\frac{s_M - \overline{s}}{\tilde{\sigma}_s}\right)^2\right) \tag{5.16}$$

where

- $\overline{s}$ is the sound pressure measurement predicted by the track $\overline{\tau}$

- $\tilde{\sigma}_s^2 = \overline{\sigma}_s^2 + \sigma_s^2$ is the corresponding innovations variance.

$\overline{\sigma}_s^2$ is the sound presure measurement prediction by track $\overline{\tau}$ while σ_s^2 is a predetermined artificial sound pressure measurement error variance.

The likelihoods (5.14) and (5.16) are commonly used in many multitarget tracking algorithms and become extremely small if $\mid \phi_M - \overline{\phi} \mid$ or $\mid s_M - \overline{s} \mid$ is large. In such a case, the likelihood is set to be zero rather than a very small but still positive value. This is done by thresholding as $\mid \phi_M - \overline{\phi} \mid / \tilde{\sigma}_\phi < \varsigma_\phi$ and $\mid s_M - \overline{s} \mid / \tilde{\sigma}_s < \varsigma_s$ with appropriate thresholding levels ς_ϕ and ς_s. Such levels may be determined from the χ^2 table.

Case 2: Merged Measurement. In order to calculate the likelihood function in case of $\#(\overline{\mathbf{T}})=2$, i.e., when measurement merging occurs, we must make some approximations. First we approximate q in (5.6) by

$$\overline{q} = \begin{cases} 1 & \text{if } \overline{s}_1 > 5\overline{s}_2 \\ \dfrac{\overline{s}_1}{\overline{s}_1 + \overline{s}_2} & \text{otherwise} \end{cases} \tag{5.6'}$$

where $\overline{s}_1$ and $\overline{s}_2$ are the sound pressure predictions of the two tracks. Likewise, we approximate (5.7) by

$$s_M^m = \begin{cases} s_M^1 & \text{if } \overline{s}_1 > 5\overline{s}_2 \\ s_M^1 + \dfrac{1}{2}s_M^2 & \text{otherwise} \end{cases} \tag{5.7'}$$

We denote the right hand side of (5.7') as $h_s^m(s_M^1, s_M^2\,;\,\overline{s}_1, \overline{s}_2)$. With these approximations, we have

$$L_{\phi}(\phi_M \mid \{\overline{\tau}_1,\overline{\tau}_2\}) = g\left(\phi_M - h_{\phi}^{m}(\overline{\phi}_1,\overline{\phi}_2\,;\,\overline{q})\,;\,\tilde{\sigma}_{\phi}^{m}\right) \qquad (5.17)$$

$$\left[\mathrm{erf}\left(\frac{\delta\phi - \tilde{\Delta}\phi}{\tilde{\sigma}_{\Delta\phi}}\right) - \mathrm{erf}\left(\frac{-\delta\phi - \tilde{\Delta}\phi}{\tilde{\sigma}_{\Delta\phi}}\right)\right]$$

where

- $h_{\varphi}^{m}(\overline{\varphi}_1,\overline{\phi}_2\,;\,\overline{q}) \triangleq \overline{q}\,\overline{\phi}_1 + (1-\overline{q})\overline{\phi}_2$

- $g(\xi\,;\,\sigma)^{\Delta}\exp(-\xi^2/2)/(\sqrt{2\pi}\sigma)$ is the probability density of a zero-mean gaussian variable

- $\mathrm{erf}(x)^{\Delta} \int_{-\infty}^{x} g(\xi)d\,\xi$ is the error function, and

-

$$\tilde{\sigma}_{\varphi}^{m} = \sqrt{[\overline{q}\,\sigma_w^1]^2 + [(1-\overline{q})\sigma_w^2]^2 + [\overline{\sigma}_{\phi}^1]^2 + [\overline{\sigma}_{\phi}^2]^2} \qquad (5.18)$$

with σ_w^i being the SD of the measurement noise in (5.2), and $\overline{\sigma}_{\phi}^i$ being the SD of the acoustic azimuth prediction error determined by $\overline{\tau}_i$ for each i. The other parameters are

$$\tilde{\Delta}\phi = \overline{\phi}_1 - \overline{\phi}_2 + \frac{\overline{q}\tilde{P}_1 - (1-\overline{q})\tilde{P}_2}{\overline{q}^2\tilde{P}_1 + (1-\overline{q})^2\tilde{P}_2}(\phi_M - h_{\phi}^{m}(\overline{\phi}_1,\overline{\phi}_2\,;\,\overline{q})) \quad , \qquad (5.19)$$

and

$$\tilde{\sigma}_{\Delta\phi} = \sqrt{\frac{\tilde{P}_1\tilde{P}_2}{\overline{q}^2\tilde{P}_1 + (1-\overline{q})^2\tilde{P}_2}} \quad , \qquad (5.20)$$

where $\overline{\phi}_i$ is the acoustic azimuth prediction by the local or global GTSD component of track $\overline{\tau}_i$ and $\tilde{P}_i = [\overline{\sigma}_{\phi}^i]^2 + [\sigma_w^i]^2$, for each i.

For the sound pressure part, we have

$$L_s(s_M \mid \{\overline{\tau}_1,\overline{\tau}_2\}) = g\left(s_M - h_s^{m}(\overline{s}_1,\overline{s}_2\,;\,\overline{s}_1,\overline{s}_2)\,;\,\tilde{\sigma}_s^{m}(\overline{s}_1,\overline{s}_2)\right) \qquad (5.21)$$

where

$$\tilde{\sigma}_s^m(\bar{s}_1,\bar{s}_2) = \sqrt{[\sigma_s^m(\bar{s}_1,\bar{s}_2)]^2 + [\bar{\sigma}_s^1]^2 + [\bar{\sigma}_s^2]^2} \quad , \tag{5.22}$$

$\bar{\sigma}_s^1$ is the SD of the sound pressure prediction error determined by the SPTSD component of track $\bar{\tau}_i$ for each i, and $\sigma_s^m(\bar{s}_1,\bar{s}_2)$ is either $\sqrt{5/2}\sigma_w^s$ or σ_w^s, depending on the condition in (5.7'), with σ_w^s being the SD of the noise term in (5.8). The derivation of (5.17) and (5.21) is described in the appendix.

Tracks Not Merging Likelihood. $L(\theta \mid \{\bar{\tau}_1,\bar{\tau}_2\})$ is the likelihood (probability) of tracks, $\bar{\tau}_1$ and $\bar{\tau}_2$, not being merged, i.e.,

$$L(\theta \mid \{\bar{\tau}_1,\bar{\tau}_2\}) = 1 - Prob.\left\{\{\bar{\tau}_1,\bar{\tau}_2\}\right\} \tag{5.23}$$

with

$$Prob.\left\{\{\bar{\tau}_1,\bar{\tau}_2\}\right\} = \mathrm{erf}\left(\frac{\delta\phi - \overline{\Delta\phi}}{\sqrt{\bar{P}_1 + \bar{P}_2}}\right) - \mathrm{erf}\left(\frac{-\delta\phi - \overline{\Delta\phi}}{\sqrt{\bar{P}_1 + \bar{P}_2}}\right) \tag{5.24}$$

being the probability of the two tracks being merged, where $\overline{\Delta\phi} = \bar{\phi}_1 - \bar{\phi}_2$.

Missed Target Likelihood. The target detection model yields likelihood (probability) of a target hypothesized by the track $\bar{\tau}$ being undetected in the current scan, i.e.,

$$L(\theta \mid \{\bar{\tau}\}) = 1 - \mathrm{erf}\left(\frac{s_{TH} - \bar{s}}{\bar{\sigma}_s}\right) \tag{5.25}$$

Thus the evaluation of the newly expanded hypotheses is equivalent to the calculation of all the likelihood functions defined above. Therefore, it is convenient to store all the above likelihoods in a table. We call such a table an *extended (because it includes merged measurements) track-to-measurement cross-reference table.*

Parallel to the calculation of each likelihood, we can update each track according to the assumed measurement assignment. When a measurement is assigned to a single track, both the GTSD and SPTSD components of the track can be updated by the Kalman filter or the extended Kalman filter. The latter filter is used for the global GTSD component. The necessary partial derivative calculation can be found in [14]. When a measurement is assigned to two merged tracks, using the approximate joint measurement equations, Equation (5.5) with q being replaced by $\bar{q}$ (for the GTSD component) and Equation (5.7') (for the SPTSD component), the GTSD and SPTSD components can be jointly updated. The resulting cross-correlation between two tracks is then ignored for simplicity.

When no measurement is assigned to a track, the TSD components are not updated. When a measurement is assigned to the null track, i.e., a single measurement is used to initiate a new track, a single-term local GTSD component and a SPTSD component are generated using the appropriate variance matrices.

5.2.3.3 Hypothesis Management

Updated clusters are then subject to *hypothesis management* operations including 1) hypothesis pruning in which low-probability hypotheses are cut off, 2) hypothesis combining in which similar hypotheses are combined, and 3) cluster splitting in which confirmed or nearly confirmed tracks are split from a cluster.

5.2.4 Information Fusion

When hypotheses are received from other nodes, they are fused with the hypotheses at the node to form new hypotheses. As discussed in Section 2, the basic steps include hypothesis formation, evaluation and management. The distributed nature of the processing necessitates operations for checking that only consistent hypotheses are formed and removing redundant information in hypothesis evaluation. These operations are facilitated by means of the information graph (also discussed in Section 2) which is an abstract model of the communication and processing in the DSN. In the following discussion, we use the terms *home* and *foreign* to represent the information present the node and that coming in from an external node.

Although we discuss hypothesis formation and evaluation separately, in actual implementation, they are usually performed simultaneously so that no unnecessary hypothesis expansion is included. For example, it is possible that a hypothesis pair $(\bar{\lambda}_1, \bar{\lambda}_2)$ satisfies the necessary condition for the fusability but yields zero probability.

5.2.4.1 Hypothesis Formation

The key problem in hypothesis formation is in identifying the fusable hypotheses and tracks from the home and foreign hypotheses and tracks. According to Section 2, the entire fusion problem can be defined in terms of the information graph. Both the home and foreign information states (tracks and hypotheses, etc.) are defined at information nodes i_1 and i_2. Then consistency checking in hypothesis formation starts by finding the *minimum set* of common predecessors of i_1 and i_2 in the information graph. By tracing back the graph to this minimum set, fusability can be determined.

For two tracks, a home track $\bar{\tau}_1$ and a foreign track $\bar{\tau}_2$, are fused whenever they are fusable. The two tracks are fusable if and only if they share the same predecessor track on each information node in the common predecessor set.

5.2.4.2 Hypothesis Evaluation

Each fused hypothesis is evaluated using equation (2.28) of Section 2. Let λ be the fused hypothesis and Z be the cumulative data at the fusion information node, then

$$P(\lambda \mid Z) = C^{-1} \prod_{i \in I_R} P(\lambda_{\mid i} \mid Z_i)^{\alpha(i)} \prod_{\tau \in \lambda} l(\tau) \qquad (5.26)$$

where C is the normalizing constant, (I_R, α) is the *information redundancy indicator*, $\lambda_{\mid i}$ is the predecessor of λ on the information node i, and $l(\tau) = L(\bar{\tau}_1, \bar{\tau}_2)$ with $(\bar{\tau}_1, \bar{\tau}_2)$ being the pair of tracks uniquely determined by a fused track τ. (I_R, α) has been defined in Section 2 (where it is denoted as $(\bar{I}, \alpha)$) and represents the redundant information at the two information nodes i_1 and i_2. I_R is the set of information nodes which affect the common information and α, a function which takes on value of +1 or -1, indicates how the redundant information can be removed.

A key step in hypothesis evaluation is the computation of the track-to-track likelihood $L(\bar{\tau}_1, \bar{\tau}_2)$ for every fusable pair $(\bar{\tau}_1, \bar{\tau}_2)$ of home and foreign tracks. For each of the tracks in the given pair, the last time when the track was updated is examined. If the updating times are different, the TSD of the track which has not been recently updated is extrapolated so that the two TSD's correspond to the target state at the same time. Then the track-to-track likelihood is calculated from the GTSD factors of the two tracks.

Whenever the likelihood is positive, the fused track $\tau = \bar{\tau}_1 \cup \bar{\tau}_2$ is created. Each fused track τ is then associated with a fused TSD (target state distribution) which is created by fusing the TSD's of the tracks from which it is fused. The GTSD component for the fused track is created from the GTSD components of the tracks from which it is fused. The SPTSD component of the fused track is the same as that of the home track in the track pair to be fused.

A track is called *local* if it consists of measurements from only one DSN sensor node; otherwise it is *global*. A global track always has a global GTSD component (i.e., a geolocational distribution in the global cartesian coordinate). A local track usually only has a local GTSD component (i.e., a geolocational distribution on the acoustic azimuth and its derivative(s)). The SPTSD component is always local.

Since the home and foreign tracks may be local or global or even empty, the computation of the track-to-track likelihoods has to consider all these possibilities. The different types of track-to-track likelihoods are shown in Figure 5-7. The calculation of the track-to-track likelihood and the fused GTSD component for each fused track is described in the following subsections for all possible combinations of home and foreign GTSD components. Because of symmetry, some of the combinations are the same. Note also that we have ignored the case of empty home track and local foreign track since the DSN node would not know how to use the azimuth information coming it. The track-to-track likelihood is thus set to zero. To simplify the notation, we assume that each GTSD component only has a single term. The results can be generalized to the case of sum-of-gaussians.

FOREIGN TRACKS

		EMPTY	GLOBAL	LOCAL
HOME TRACKS	EMPTY	1	4	IGNORE
	GLOBAL	4	2	3
	LOCAL	6	3	5

Figure 5-7: Possible Track-to-Track Combinations

5.2.4.2.1 CASE 1: Empty Home Track/Empty Foreign Track

The density β_{ND}^{i} of the undetected targets at each information node i in the set I_R is either stored in the current information node before the current message is received or contained in the received message. The updated density β_{ND} for the current information node is calculated as

$$\beta_{ND} = L(\emptyset,\emptyset) = \overline{M}(\emptyset,\emptyset) = \prod_{i \in I_R} (\beta_{ND}^{i})^{\alpha(i)} \tag{5.27}$$

where (I_R, α) is the information redundancy indicator and $\overline{M}(\cdot,\cdot)$ is defined as

$$\overline{M}(\overline{\tau}_1,\overline{\tau}_2) = \prod\{(\beta_{ND}^{i})^{\alpha(i)} \mid i \in I_R \;, (\overline{\tau}_1 \cup \overline{\tau}_2)\mid_i = \emptyset\} \tag{5.28}$$

with $\tau|_i$ is the restriction of a track τ to an information node i, i.e., $\tau|_i = \tau \cap J_i$ where J_i is the cumulative measurement index set at the information node i.

5.2.4.2.2 CASE 2: Global Home Track/Global Foreign Track

Suppose both the home and foreign tracks τ_1 and τ_2 are global. The information nodes in the set I_R consists of two types:

- Those where the common predecessor of τ_1 and τ_2 have a global GTSD factor

- Those where the common predecessor of τ_1 and τ_2 have a local GTSD factor

Let

$$I_R^G = \{i \in I_R \mid (\tau_1 \cup \tau_2)\mid_i \text{ has global GTSD component}\} \tag{5.29}$$

The predecessor track at each $i \in I_R^G$ has a global GTSD component (position and velocity) with mean $\hat{x}_i$ and variance matrix Σ_i. Then the part of the track-to-track likelihood concerning with I_R^G is given by

$$L_G(\tau_1,\tau_2) = \left(\frac{\prod_{i \in I_R^G} \det(\Sigma_i)}{\det(\Sigma)}\right)^{\frac{1}{2}} \exp\left(-\frac{1}{2} \sum_{i \in I_R^G} \|\hat{x} - \hat{x}_i\|^2_{\Sigma_i^{-1}}\right) \tag{5.30}$$

The mean $\hat{x}$ and variance Σ of the (global) GTSD component of the fused track $\tau_1 \cup \tau_2$ are given as

$$\hat{x} = \Sigma \sum_{i \in I_R^G} \alpha(i) \Sigma_i^{-1} x_i \tag{5.31}$$

and

$$\Sigma = \Big(\sum_{i \in I_R^G} \alpha(i) \Sigma_i^{-1} \Big)^{-1} \tag{5.32}$$

In Equations (5.28) - (5.31), (I_R, α) is the information redundancy indicator. Let $I_R^L = I_R \setminus I_R^G$, i.e., the set of common predecessor nodes where the tracks have local GTSD components. For each i in I_R^L, define

$$p_i(\tau_1, \tau_2) = p_i^L(\hat{\phi}, \hat{\dot{\phi}})\ \bar{p}^V(\hat{V}, \hat{\psi})\ J(\hat{x}) \tag{5.33}$$

where

- $(\hat{\phi}, \hat{\dot{\phi}})$ is the pair of the estimates of the acoustic azimuth and its derivative calculated from $\hat{x}$,

- $(\hat{V}, \hat{\psi})$ is the target speed/heading estimated by $\hat{x}$,

- $p_i^L(\cdot,\cdot)$ is the density of the GTSD component of the predecessor track at i (marginal to $(\phi, \dot{\phi})$),

- $\bar{p}^V(\cdot,\cdot)$ is the density of the *a priori* distribution of the target speed and the heading, and

- $J(\cdot)$ is the appropriate Jacobian.

Define

$$L_L(\tau_1, \tau_2) = \prod_{i \in I_R^L} p_i(\tau_1, \tau_2)^{\alpha(i)} \tag{5.34}$$

Then the track-to-track likelihood is calculated as

$$L(\tau_1, \tau_2) = \bar{M}(\tau_1, \tau_2)\ L_L(\tau_1, \tau_2)\ L_G(\tau_1, \tau_2) \tag{5.35}$$

When the GTSD component of the fused track is multiple-termed, (5.15) is calculated for each term and the weighted sum becomes $L_L(\tau_1, \tau_2)$ with the new weights for the fused track. The SPTSD component of the fused track is identical to that of the home track.

5.2.4.2.3 CASE 3: Global Home Track/Local Foreign Track

When the foreign track τ_2 is local, every predecessor track $(\tau_1 \cup \tau_2)|_i$ is empty for every $i \in I_R$ except for i_1 and i_2. If the foreign track has a global GTSD, the calculation of the likelihood and the fused GTSD can be done as in CASE 1.

Suppose the foreign track has a local GTSD component. Then the GTSD component of the fused track $\tau_1 \cup \tau_2$ has the mean $\hat{x}$ and variance matrix Σ, which are calculated by the extended Kalman filter equations:

$$\hat{x} = \hat{x}_1 + K(\hat{\Phi}_2 - \bar{\Phi}_1) \tag{5.36}$$

where

- $\hat{x}_1$ is the mean of the GTSD of the home track τ_1,

- $\hat{\Phi}_2$ is the vector of the means of the acoustic azimuth and its derivative in the GTSD component of the foreign track τ_2,

- $\bar{\Phi}_1$ is the azimuth and its derivative of the target at i predicted by the $\hat{x}_1$.

K in (5.36) is the filter gain defined by

$$K = \bar{\Sigma}_1 H^T S^{-1} \tag{5.37}$$

where

-

$$S = H \bar{\Sigma}_1 H^T + R \quad , \tag{5.38}$$

- $\bar{\Sigma}_1$ is the variance matrix of the GTSD component of the home track τ_1,

- R is the variance submatrix of the local GTSD component of the foreign track τ_2, and

- H is the derivative of the transformation function h which transforms the global target state into the local coordinates used for the GTSD component of the foreign track.

The variance Σ of the fused track is then given by

$$\Sigma = (I - KH)\bar{\Sigma}_1 \tag{5.39}$$

The track-to-track likelihood is calculated as

$$L(\tau_1,\tau_2) = \bar{M}(\tau_1,\tau_2)\, L_{GL}(\tau_1,\tau_2) \tag{5.40}$$

where

$$L_{GL}(\tau_1,\tau_2) = (2\pi)^{-1}(\det(S))^{\frac{1}{2}} \exp\left(-\frac{1}{2}\|\hat{\Phi}-\bar{\Phi}\|_{s^{-1}}\right) \bar{p}^{V}(\hat{V},\hat{\psi})\, J(\hat{x}) \tag{5.41}$$

where

- $\bar{p}^{V}(\cdot,\cdot)$ is the density of the *a priori* distribution on the target speed and the heading,

- $J(\cdot)$ is the Jacobian and $(\hat{V},\hat{\phi})$ is the the target speed and the heading estimated by $\hat{x}$.

5.2.4.2.4 CASE 4: Global Home Track/Empty Foreign Track

When the foreign track τ_2 is empty, all the predecessor track $(\tau_1 \cup \tau_2)|_i$ is empty except for i_1. Therefore, we have $L(\tau_1,\tau_2)=\bar{M}(\tau_1,\tau_2)$ and the TSD of the home track becomes the TSD of the fused track.

5.?.4.2.5 CASE 5: Local Home Track/Local Foreign Track

This is the case when two local tracks from two sensor nodes are used to initiate a global track. When the home track τ_1 and foreign track τ_2 are both local $(\tau_1 \cup \tau_2)|_i = \emptyset$ except for i_1 and i_2. The fused GTSD component is created first. This is done by using the "position" track initiation equation described in [15]. As before, we assume that both home and foreign tracks have single-termed GTSD components. Then, using the means and the variances of the of the two local azimuth values and their first-order derivatives, the global GTSD distribution is obtained by solving a multidimensional algebraic equation. The algebraic equation is quadratic and may not have any solution. In such a case, the track-to-track likelihood is zero and the fused track is not created. Otherwise we have two GTSD components with means and variances, $(\hat{x}_1,\Sigma_1)$ and $(\hat{x}_2,\Sigma_2)$, corresponding to the two solutions to the algebraic equation.

Then, for each $k \in \{1,2\}$, we calculate

$$a_k = (\det(\Sigma_k))^{-\frac{1}{2}} \tag{5.42}$$

and

$$b_k = (\det(\Sigma_{\phi_1}))^{-\frac{1}{2}} (\det(\Sigma_{\phi_2}))^{-\frac{1}{2}} \bar{p}^V(\hat{V}_k, \hat{\psi}_k)^2 J(\hat{x}_k) \tag{5.43}$$

where

- Σ_{ϕ_1} and Σ_{ϕ_2} are the variance of the local (azimuth, its derivative) vector attached to the home and the foreign tracks,

- $(\hat{V}_k, \hat{\psi}_k)$ is the target speed and heading estimated by $\hat{x}_k$,

- $\bar{p}^V(\cdot,\cdot)$ is the density of the *a priori* distribution of the target velocity vector, and

- J is the appropriate Jacobian

The weights w_1 and w_2 are then calculated by

$$\frac{w_1}{w_2} = \frac{b_1}{b_2}\frac{a_2}{a_1} \tag{5.44}$$

with $w_1 + w_2 = 1$. The GTSD component of the fused track $\tau_1 \cup \tau_2$ is given as a sum-of-gaussian distribution with weights w_1 and w_2. The track-to-track likelihood is calculated as

$$L(\tau_1,\tau_2) = \overline{M}(\tau_1,\tau_2)\,\frac{b_1 + b_2}{a_1 w_1 + a_2 w_2} \tag{5.45}$$

5.2.4.2.6 CASE 6: Local Home Track/Empty Foreign Track

As in CASE 4, $L(\tau_1,\emptyset)=\overline{M}(\tau_1,\emptyset)$. The TSD of the fused track is identical to that of the home track.

5.2.4.3 Hypothesis Management

The hypothesis management procedures used in the information fusion process are almost identical to those used in the local data process, and include hypothesis pruning, hypothesis combining and clustering.

5.3 CONCLUSION

In this section, we have presented acoustic tracking scenarios which can be used to evaluate the multiple hypothesis approach to distributed tracking algorithms. These scenarios were derived from the experimental set-up at the Lincoln Lab. DSN test bed so that the experiments can be performed on the actual testbed if resources permit. The general algorithms of Section 2 have been adapted to handle acoustic sensors. Because of the characteristics of acoustic sensors, such as azimuth-only measurements and propagation delays, special techniques have been developed for hypothesis formation and evaluation, especially at the track level. However, the overall architecture of Section 2 still applies.

6. SIMULATION EXAMPLES

The complexity of the algorithms in a DSN precludes analytic approaches to performance evaluation. Thus we have developed a simulation environment as a research tool for developing algorithms, evaluating their performance, and understanding the general issues associated with a DSN. In this section, we give a brief description of the simulation environment and present some examples simulated in this environment.

6.1 SIMULATION ENVIRONMENT

We first present the hardware and software used in the simulation environment. This will be followed by a description of the user interfaces and capabilities of the software.

6.1.1 Hardware and Software

The current DSN code was first implemented on a VAX 11-780 under the UNIX operating system. It was then moved to the Symbolics 3600 Lisp Machine. LISP was chosen since the data structures for hypotheses and tracks can be represented conveniently in the form of property lists or def-structs in LISP. Furthermore, since the size of the data structure is dynamic, being driven by the sensor data and communication, efficient memory allocation and deallocation are desirable. Garbage collection is automatic in LISP, thus simplifying the coding task.

The LISP machine also provides a good environment for program development, including the use of multiple windows and utilities to support coding in LISP. It also allows graphical displays for hypotheses, and with the use of a second monitor, displays for the target movements.

In the interim report [4], we presented an architecture for a general test bed environment within which a DSN system may be designed and prototyped. This architecture, called *Schemer*, has since evolved into a programming environment called *SOPE* (System Oriented Programming Environment). A *SOPE* system is an object-oriented realization of a "system" as it is thought of in general system theory. Fundamentally, a system is an object that performs a specialized set of computations and interacts with the rest of the world sending and receiving messages. The current DSN simulation was built using some of the *SOPE* capabilities.

6.1.2 User Interface

An interface has been provided for controlling the experiment as well as displaying the simulation results to the user. This interface has also proved to be useful in program development.

Two kinds of displays have been developed for the DSN simulation: a situation display for the scenario and a hypothesis display. The situation display is on a color monitor and the hypothesis display is on a black and white monitor. Various display functions are controlled by means of menus and the mouse. The two displays are shown in Figure 6-1 and 6-2.

The hypothesis display (Figure 6-1) on the current system shows all the hypotheses for all the nodes (four at present). For each node, the evolution of the hypotheses are shown. Each circle denotes a hypothesis and the number in each hypothesis is the probability of the hypothesis. The parent and children of each hypothesis within the node are connected by lines. By pointing the mouse to a hypothesis which results from fusion, one can also identify its predecessors from other DSN nodes. In the figure, the predecessors for a fused hypothesis are darkened.

The situation display (Figure 6-2) on the current system shows the scenario under consideration, and the sensor characteristics. For each time, the target locations and the measurements can be displayed. By pointing the mouse to each hypothesis, one can also display the target locations according to the hypothesis. Thus, the evolution of the situation becomes obvious as one moves the mouse within each node. Furthermore, by considering the hypotheses from multiple DSN nodes, one can identify how information is being fused.

6.1.3 System Capabilities

The current simulation focuses on the processing within each node and a perfect (noiseless) communication model is assumed. Each node in the network is equipped with a GTC (Generalized Tracker/Classifier) which processes the local sensor data and an information fusion module which fuses the information sent from the other nodes with the local information. The current simulation has the following capabilities:

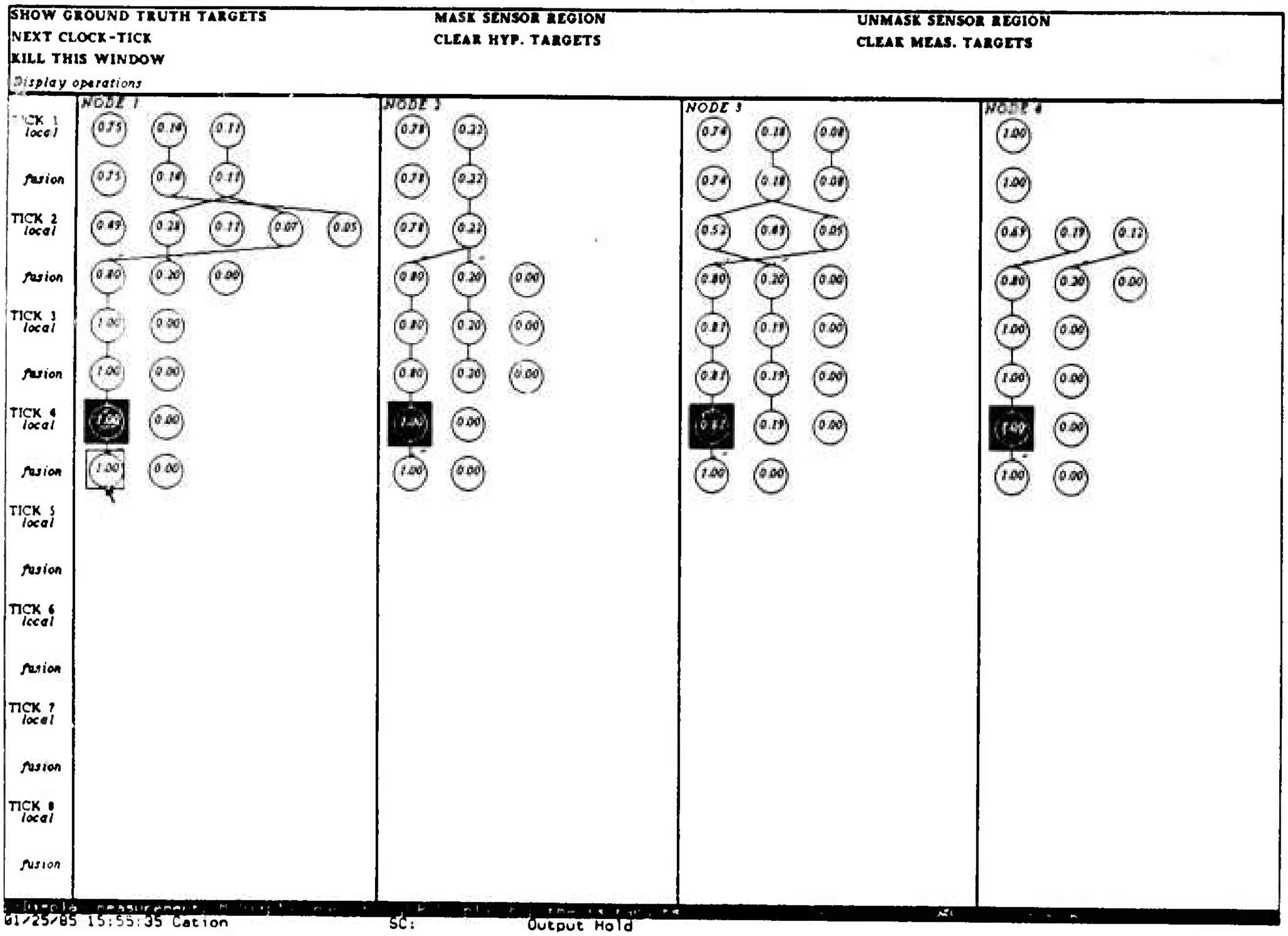


Figure 6-1: Hypothesis Display

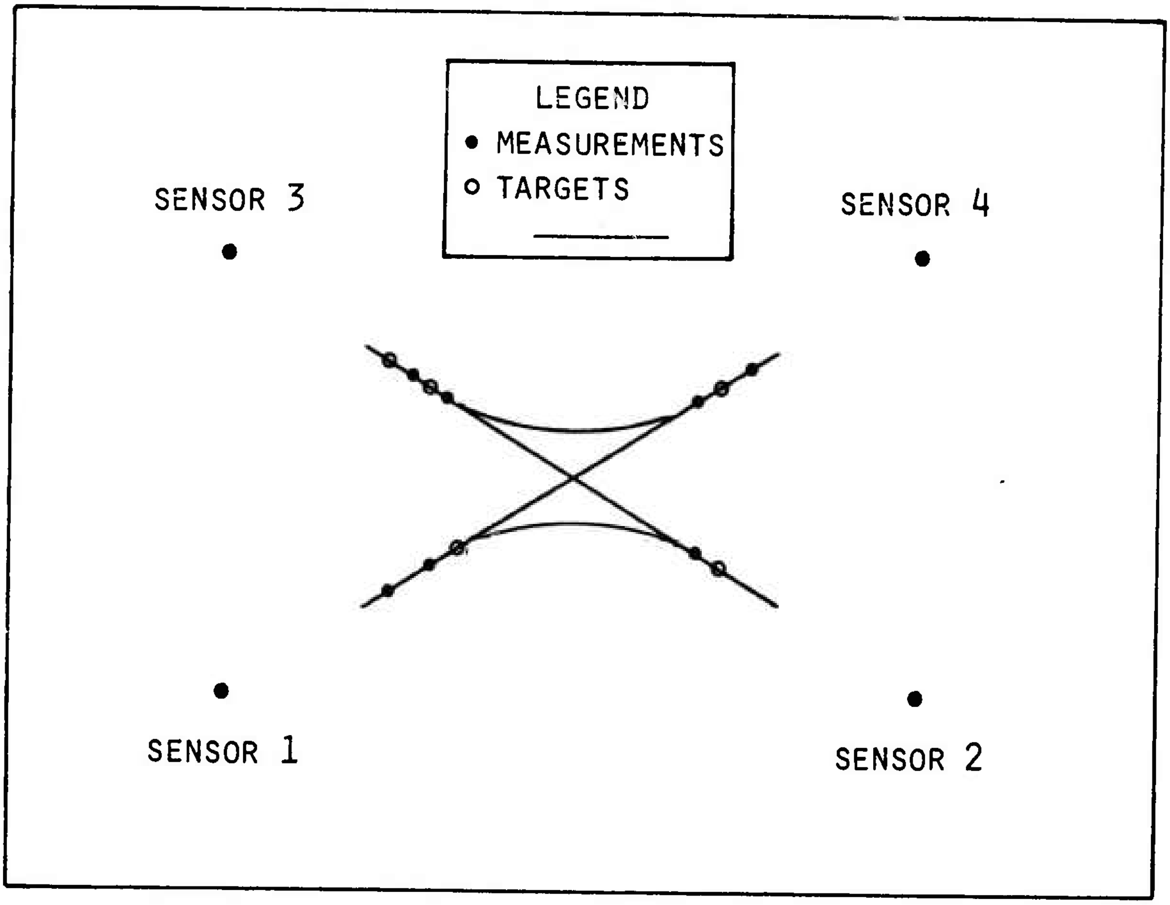


Figure 6-2: Situation Display

- The maximum number of nodes which can be handled in the network is four. There is no conceptual difficulty in increasing the limit on the number of nodes but simulation time will increase substantially since a single computer is used to simulate a distributed system.

- The communication between any two nodes can be specified arbitrarily. By using the information distribution module, adaptive time-varying communication strategies can be handled.

- The processing of the local sensor data is by means of the GTC developed in the previous project [1]. Information fusion is based on algorithms for hypothesis formation and evaluation described in Section 2.

For scenarios with average target density, false alarm rate, detection probability, and measurement accuracy, the algorithm runs reasonably fast. However, the fact that a single machine is used for data generation, communication simulation, and simulation of multiple nodes makes it difficult to evaluate the actual performance in terms of speed. Furthermore, the garbage collection of the LISP machines sometimes interferes with the processing.

In order to handle arbitrary communication patterns among the nodes, the information fusion algorithm includes mechanisms to trace the histories of the tracks and hypotheses in the information graph. Without any loss of generality, information fusion from multiple nodes is carried out sequentially in a binary form, i.e, to fuse the information from node A, B and C, we first fuse that of A and B, and then the result is fused with the information from C. This simplifies the implementation of the fusion algorithm considerably.

6.2 DISCRETE ROAD NETWORK EXAMPLE

We now present some simulation results for a four-node sensor network to illustrate the performance of the DSN fusion algorithm. We use a simple discrete-state road network scenario where the target dynamics are assumed to be Markov with the road-segments as the possible states. The main reason for using the simple target dynamics and scenario was to minimize any unnecessary numerical complexity due to target motion and to concentrate more on issues resulting from arbitrary communication pattern. The simulation program, however, is capable of handling more complicated scenarios if the appropriate algorithms are included.

6.2.1 Target and Sensor Models

The underlying models in the scenario are:

a. Targets move along the road network with discretized straight-line segments.

b. The target dynamics are Markov with a given transition matrix.

c. Each sensor measures position (segment number) along the road with some uncertainty due to the bearing and range measurement noise. Each sensor also has certain masked regions which it cannot observe.

d. The probability of detection of a target in each road-segment by a sensor is a function of sensor masking and the relative sensor location.

In addition to this, independent and identically distributed target models have also been assumed in the current simulation.

There are four nodes in the DSN, with a sensor at each node. The sensors observe the same road network although they have different fields-of-view. The road network and the location of the sensors are shown in Figure 6-3. Each individual target position is represented by the segment number and its evolution is assumed to be a Markov process. The target state at any time is thus characterized by a probability distribution on the road segment. Because of the terrain and other masking (due to foliage, etc) the sensors have masked regions. When the target moves into these regions, it will not be seen by the sensor. A sensor can fail to detect a target in the unmasked region because the probability of detection is less than one. Figures 6-4 to 6-7 show the detection probabilities of the four sensors. Each sensor generates a measurement in the following way. The detection of a target at state x_i by a sensor depends on the detection probability which is 0 whenever the target is in a masked region relative to the sensor. For any detected target located at x, the measurement y, which is also a

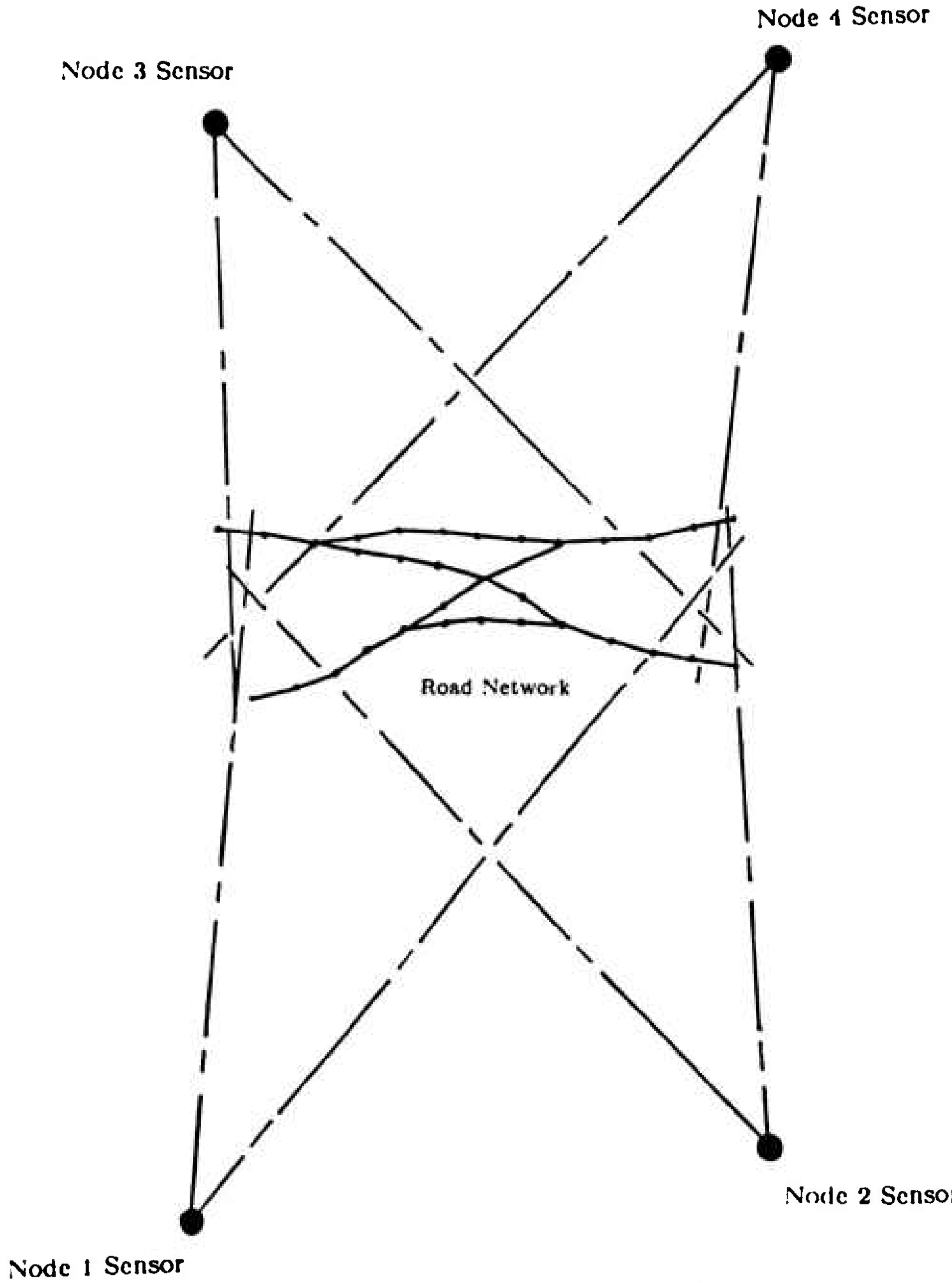


Figure 6-3: A Four Node Network Observing a Road Network

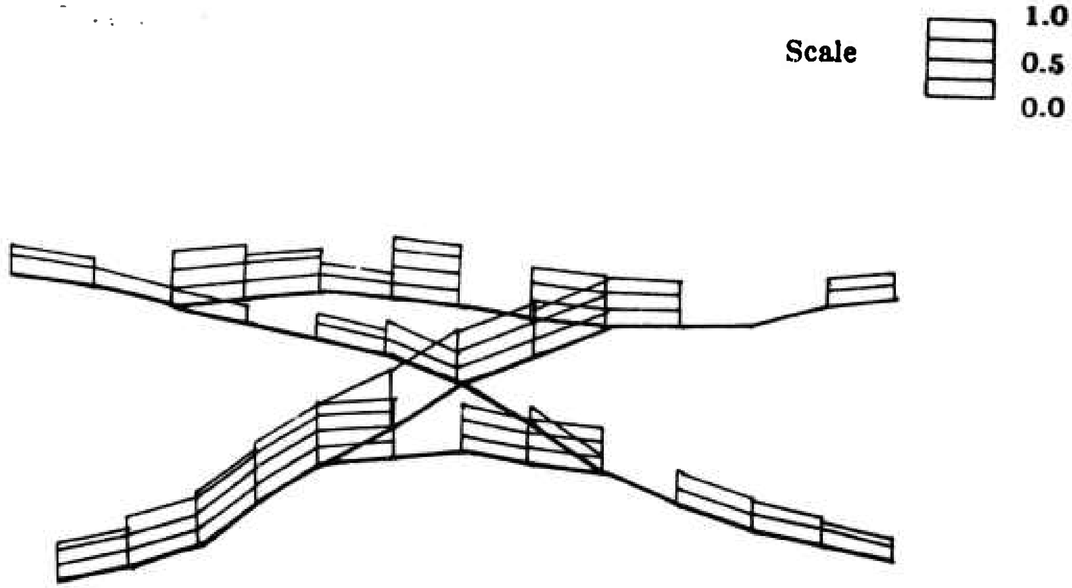


Figure 6-4: Probability of Detection of Sensor 1

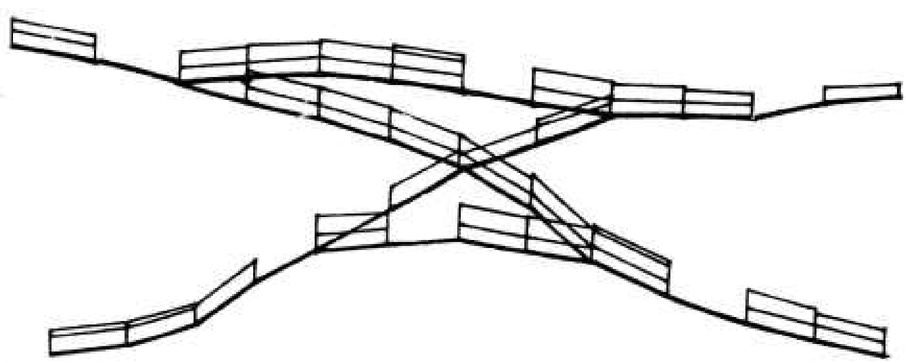


Figure 6-5: Probability of Detection of Sensor 2

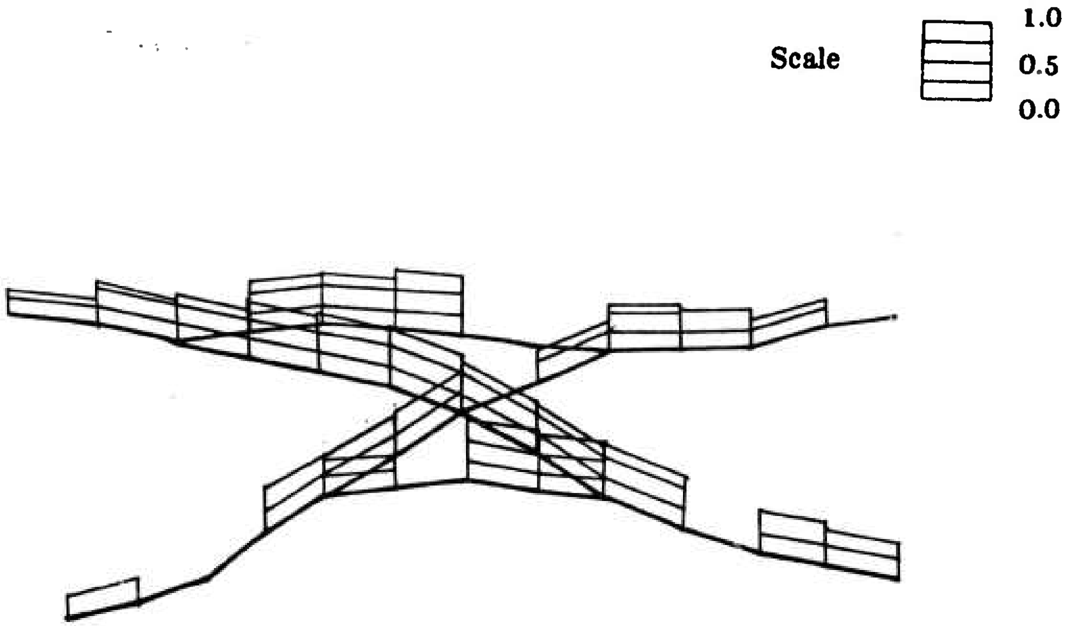


Figure 6-6: Probability of Detection of Sensor 3

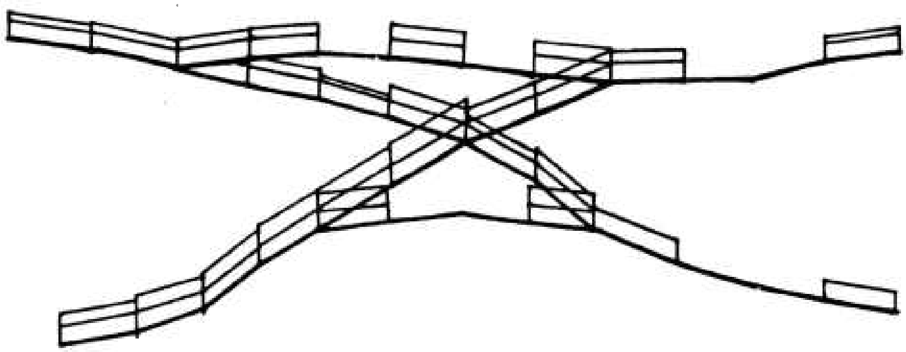


Figure 6-7: Probability of Detection of Sensor 4

segment number, is generated according to the following conditional probability distribution function: (see Figure 6-8)

$$p(y \mid x) \approx \sum_i \alpha_i(y_i)U(y_i) \tag{6.1}$$

where

$$\alpha_i(y_i) = \int_{r_{\min}(y_i)}^{r_{\max}(y_i)} \int_{\theta_{\min}(y_i)}^{\theta_{\max}(y_i)} g_r(r \mid \bar{r}(x)) g_\theta(\theta \mid \bar{\theta}(x)) dr d\theta \tag{6.2}$$

$U(y_i)$ is a uniform function on segment y_i with unity value and $g_r(r \mid \bar{r}(x))$ and $g_\theta(\theta \mid \bar{\theta}(x))$ are sensor characteristics corresponding to the the measurement uncertainty in range and bearing given the average range and bearing of a particular target location x. False alarms are also added according to the sensor model.

The total number of targets is constant but unknown and its a priori distribution is Poisson with mean ν_0. The number of false alarms in each scan is also Poisson with mean ν_{FA} for each sensor. The target positions are independent and identically distributed with the a priori distribution uniform over the road network states, and targets are expected to move into the field-of-view from the edges at any time. The parameters used in the simulations are given in Table 6-1.

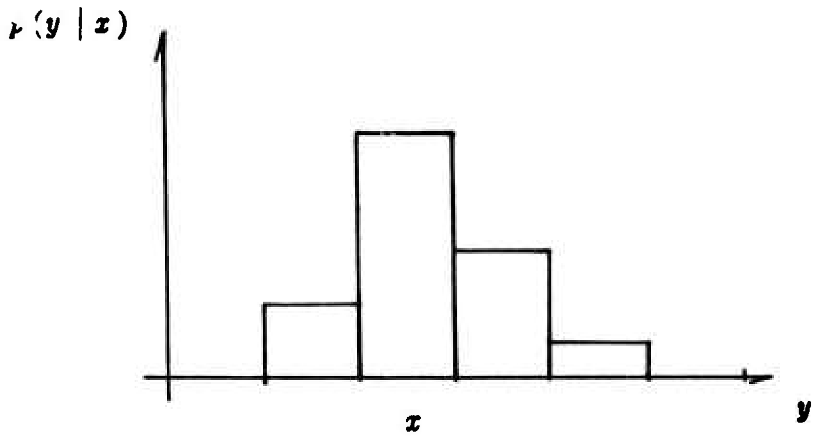


Figure 6-8: Conditional Probability Distribution $p(y|x)$

Table 6-1: Simulation Parameters

Expected number of targets		ν_0	4
Expected number of false alarm		ν_{FA}	1/scan
Probability of detection		P_D max	0.9
Measurement error	range bearing radial velocity	σ_r σ_θ σ_ρ	0.5 (km) 0.2 (radius) 0.1 (km/min)
Pruning threshold		ϵ	0.05

6.2.2 Communication Schemes

Different kinds of communication patterns were experimented. The first (decentralized case) consists of no communication among the nodes. The second is hierarchical communication with the following features:

1. At every odd scan NODE 1 sends information to NODE 2, and NODE 3 sends information to NODE 4.

2. At every even scan NODE 4 sends information to NODE 2.

i.e, *node* 1 and *node* 3 only transmit information to other nodes, *node* 4 is an intermediate receiver/processor/transmitter and all information is thus collected by *node* 2 with communication delays. The hierarchical communication pattern and the information graph are shown in Figure 6-9 and Figure 6-10. The third case considered is broadcast communication.

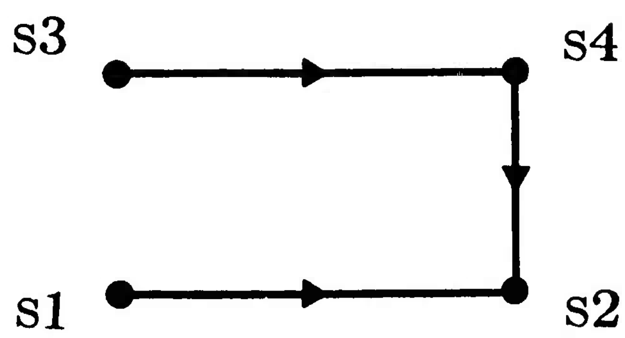


Figure 6-9: Hierarchical Communication Scheme

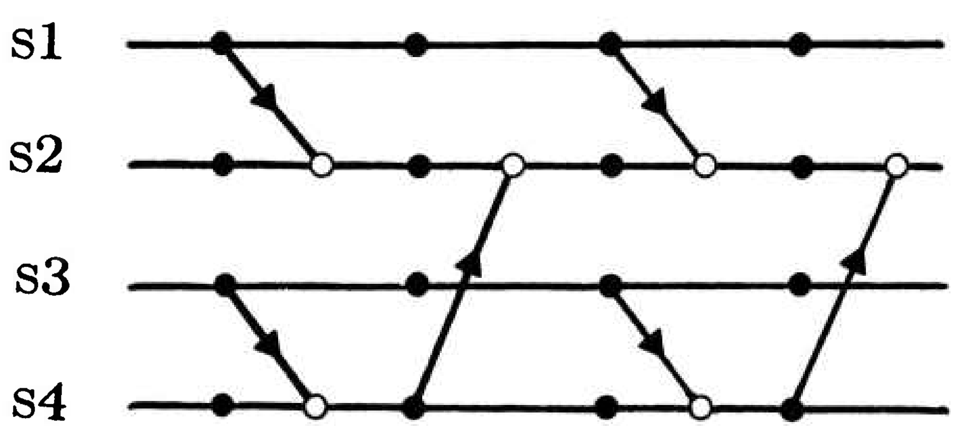


Figure 6-10: Information Graph for Hierarchical Communication

6.2.3 Simulation Results

In each simulation, all the hypotheses were examined and compared to the true trajectories of targets according to the measurement-to-target association histories. The hypothesis best matched to the ground truth is defined as a true hypothesis. The most likely hypothesis (highest probability) is called the best hypothesis.

The results of a simulation run are shown in Figures 6-11 to 6-12 where the probabilities of the true and best (with highest probability) hypotheses are plotted versus time for each of the four DSN nodes. Note how the probabilities of the hypotheses change with time. In general the true hypothesis and best hypothesis are not the same when the data quality is poor. This argues for the multiple-hypothesis approach since if only the best hypothesis is selected, an incorrect hypothesis may result. The probabilities of the best hypotheses and true hypotheses for each node for the no communication case are shown in Figure 6-11. Note that because of their data quality, the best hypotheses for *NODES* 1, 2 and 4 are not the correct hypotheses. *NODE* 3, however, tracks the targets correctly. In the hierarchical case (Figure 6-12), the hypotheses of *NODE* 1 and *NODE* 3 behave the same as in the case with no communication since they do not receive any information from other nodes *NODE* 4 now has the help of *NODE* 3 and performs much better, acquiring the correct hypothesis after a while. *NODE* 2 performs best since eventually it gets information from all nodes. In the broadcast case, the nodes all find the true hypothesis in a short time.

Figures 6-13 and 6-14 show the hypothesis trees for the two cases discussed above. The number of hypotheses for each node varies with time, depending on the complexity of the current situation. For example, in Figure 6-13, *NODE* 3 starts out with four hypotheses around scan (tick) 3, with the true hypothesis having a fairly low probability. As it collects more data, the situation clears up so that there are only two hypotheses at scan 8. This phenomenon is even more pronounced in Figure 6-14 where *NODE* 4 starts with four hypotheses with nonzero probability at scan 2 and 3. With communication from *NODE* 3, only one hypothesis has nonzero probability at scan 8.

6.3 CONTINUOUS STATE EXAMPLE

Figures 6-15 and 6-16 present some results in continuous state tracking. The network consists of two nodes (denoted by 1 and 2) which broadcast their hypotheses periodically. The azimuth measurement of each sensor is more

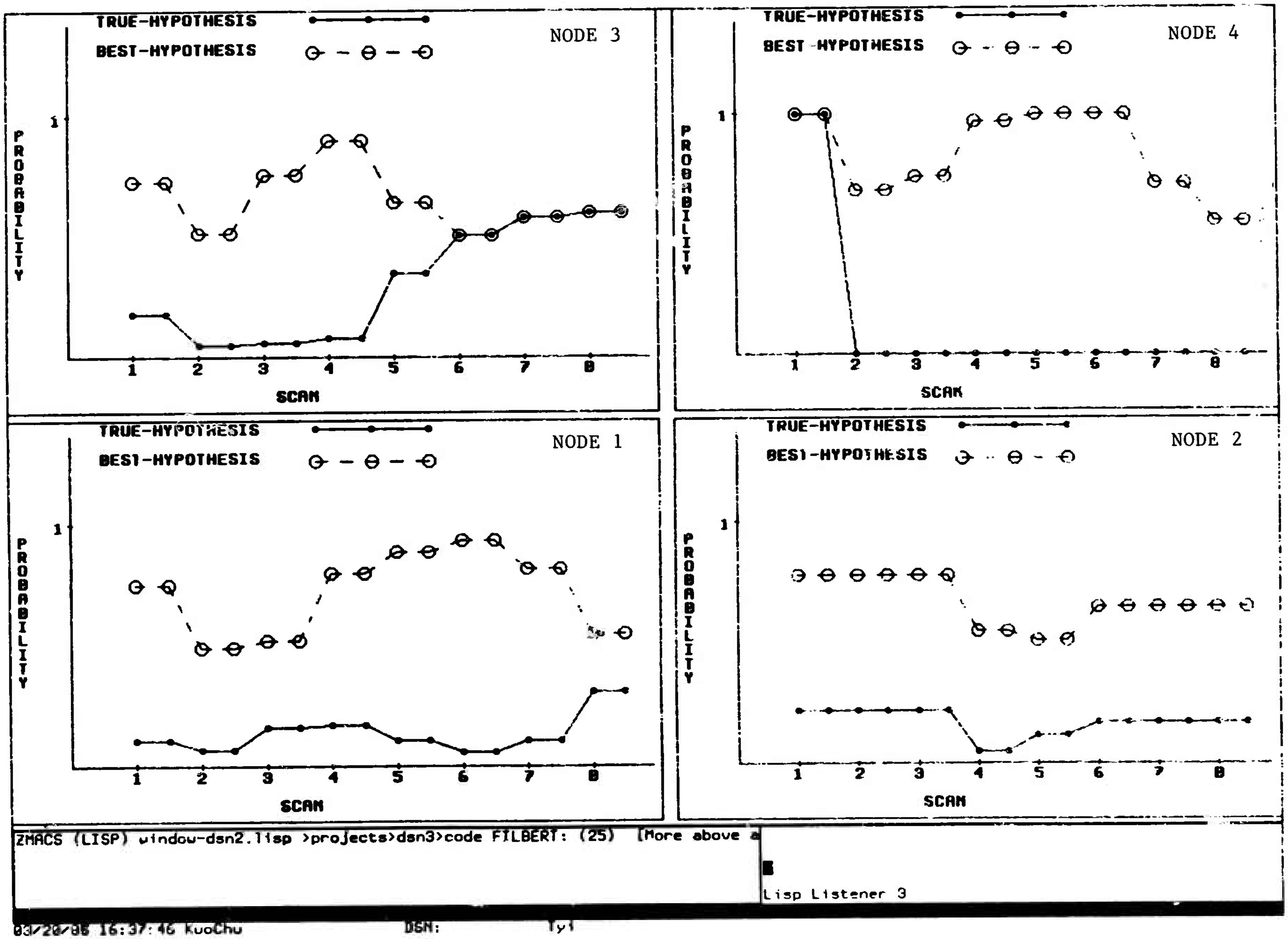


Figure 6-11: Performance of Nodes with No Communication

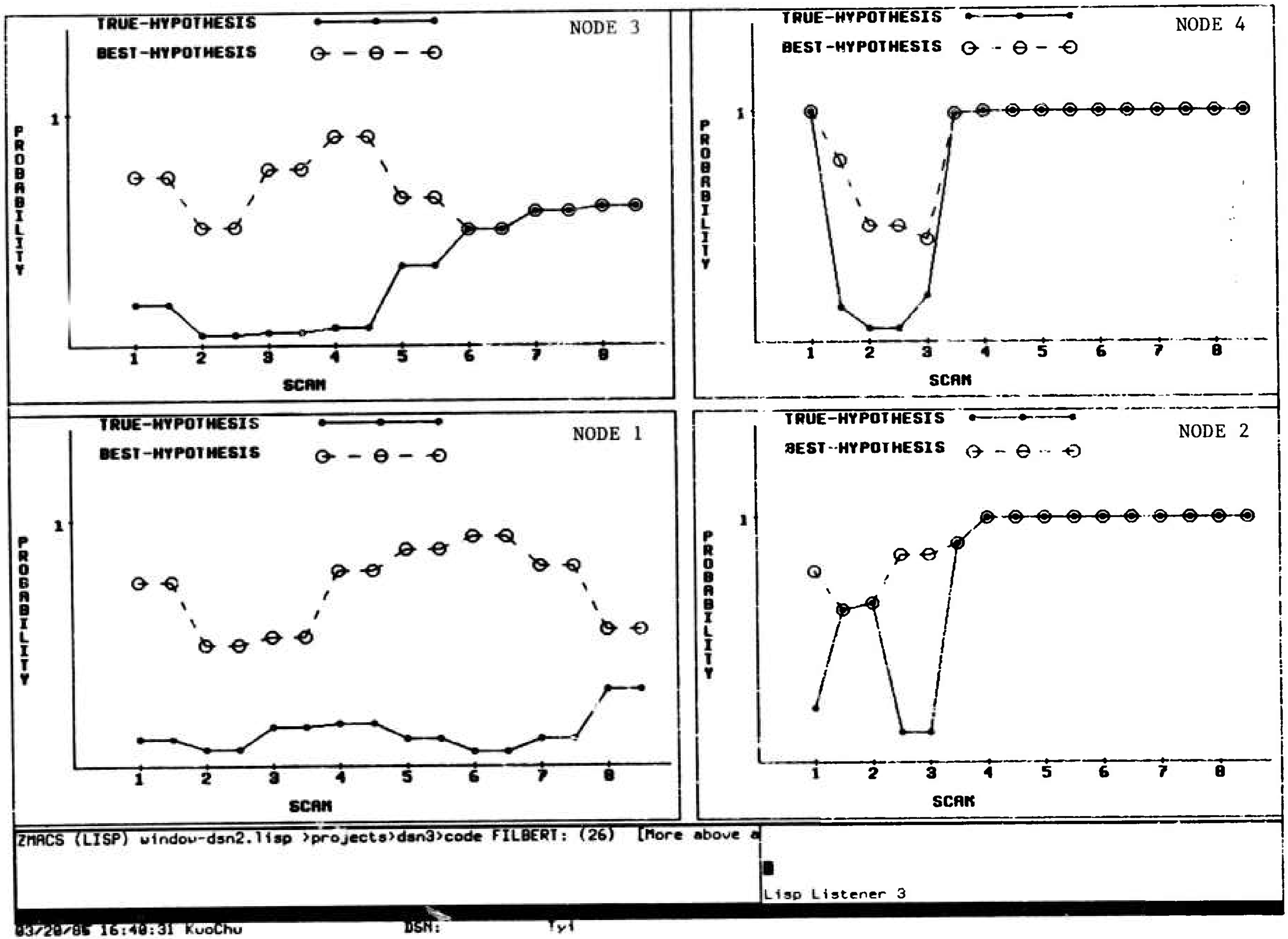


Figure 6-12: Performance of Nodes with Hierarchical Communication

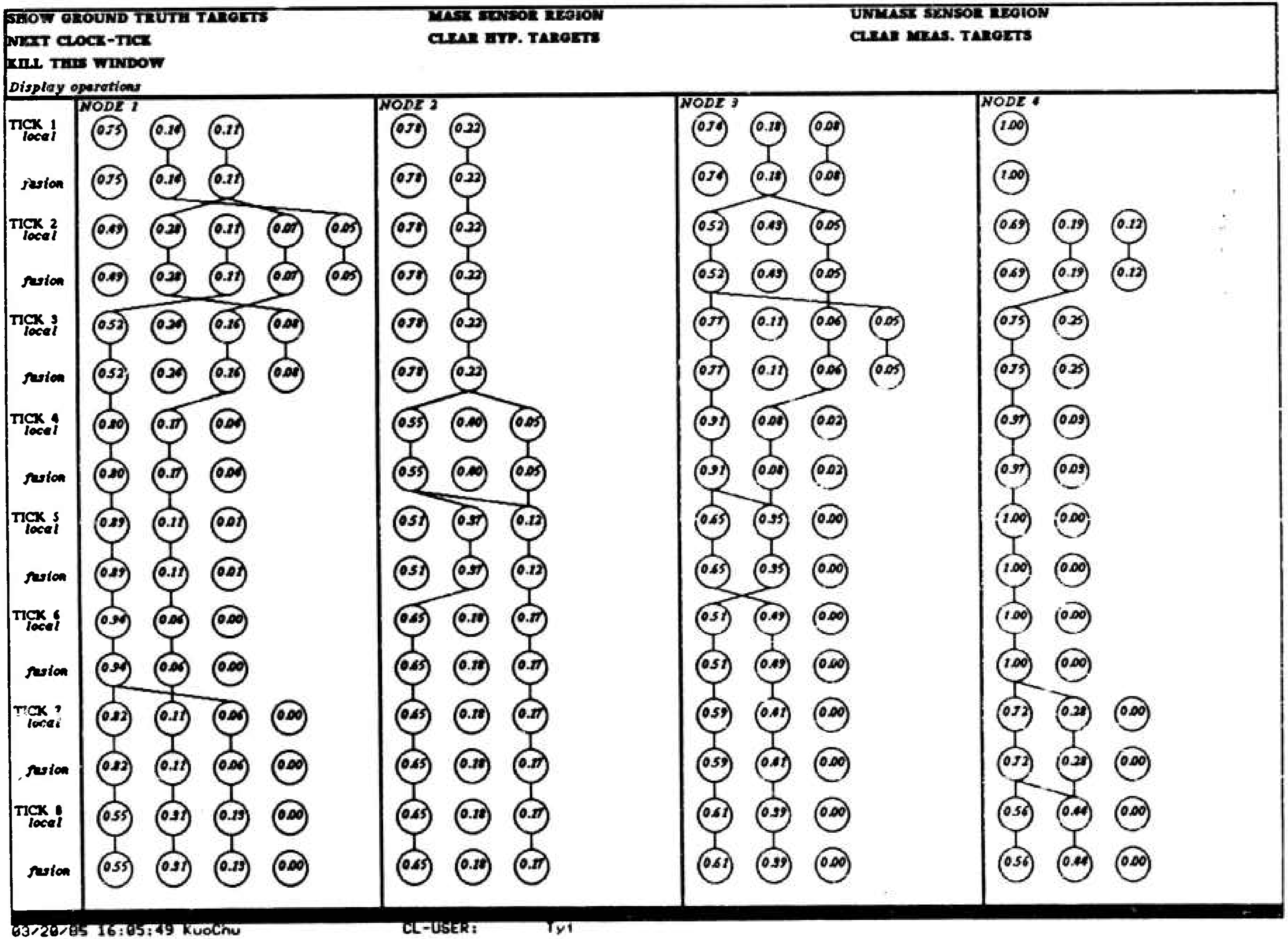


Figure 6-13: Hypothesis Trees - No Communication

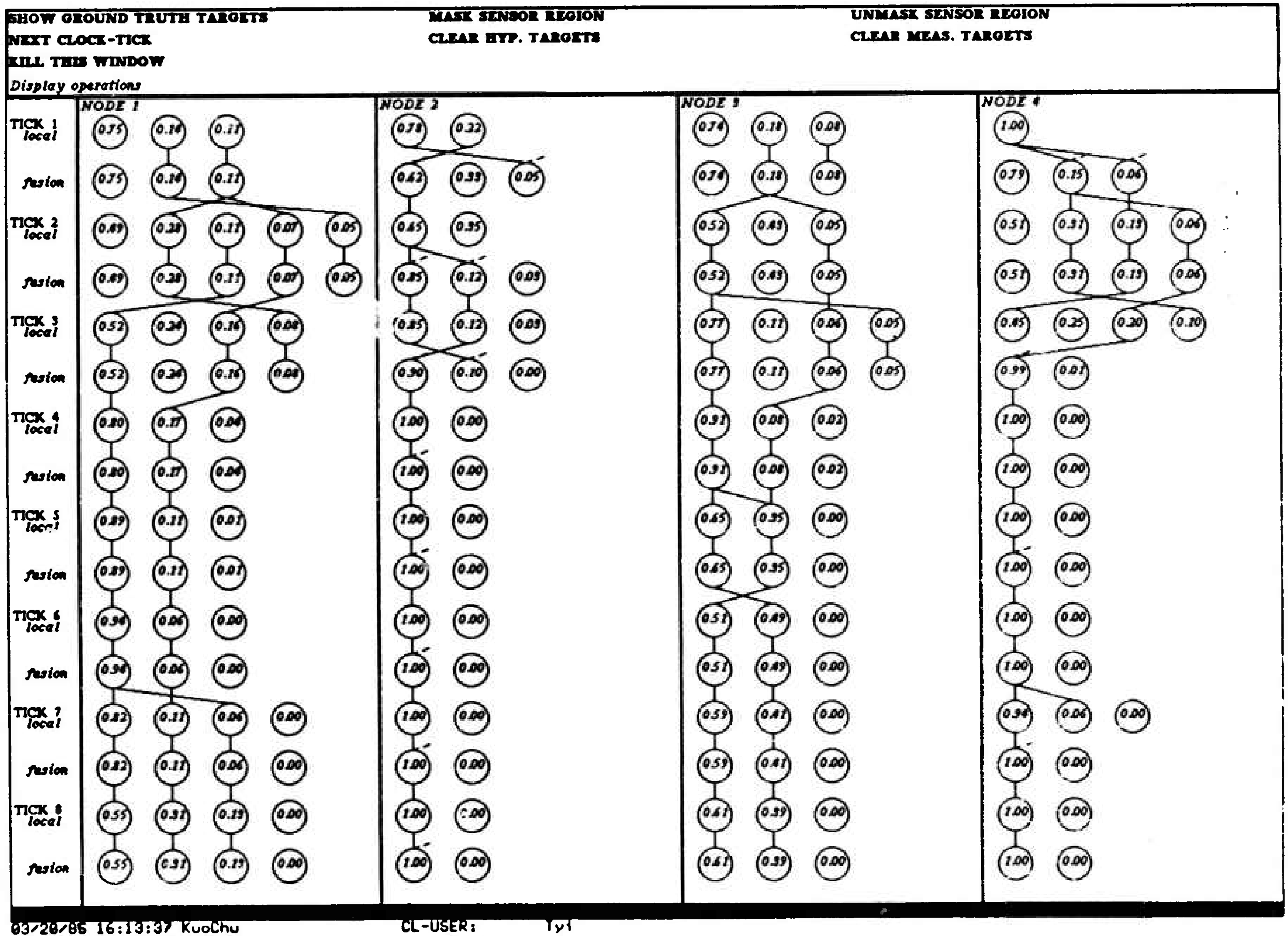


Figure 6-14: Hypothesis Trees - Hierarchical Communication

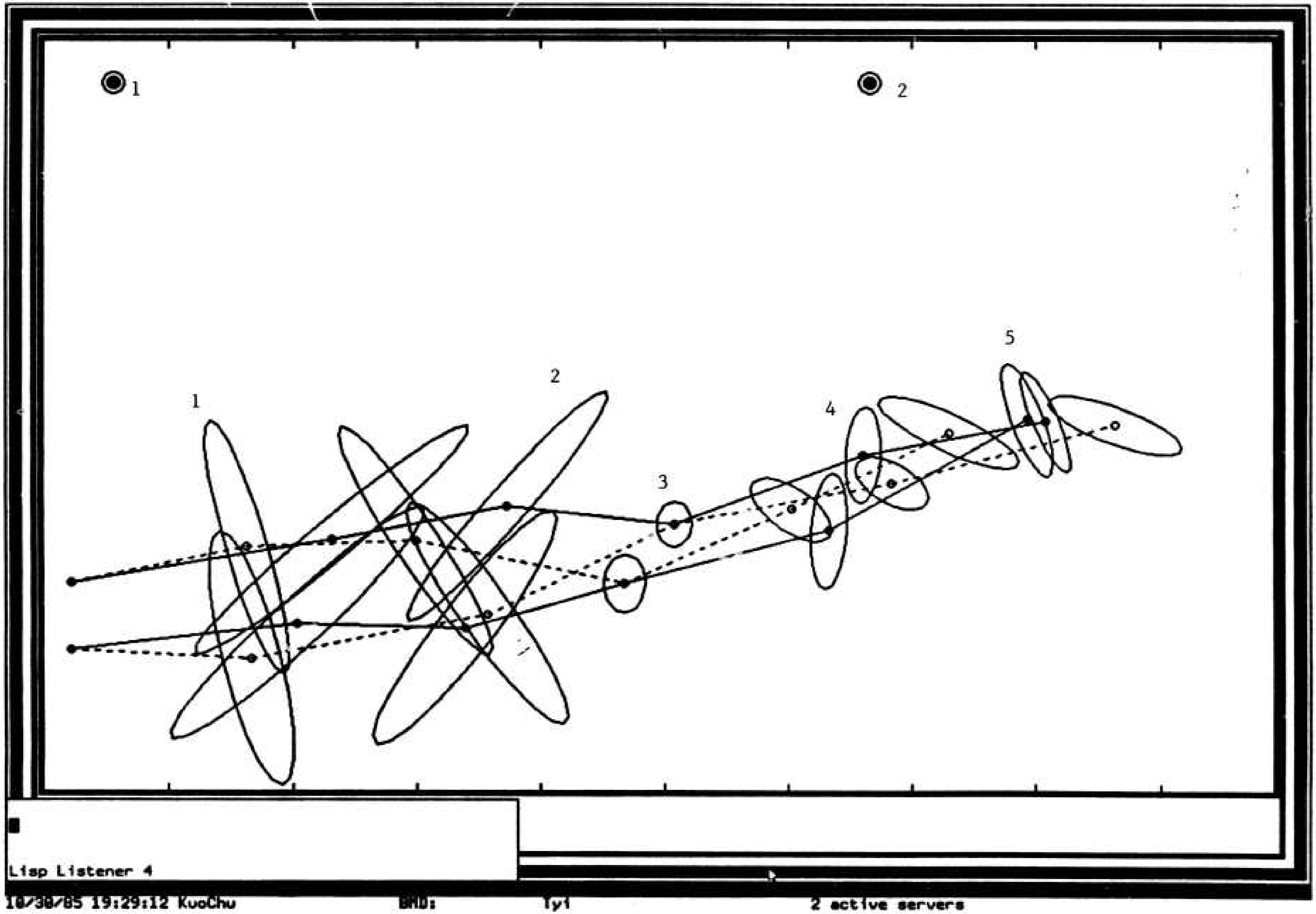


Figure 6-15: Example of Incorrect Hypothesis

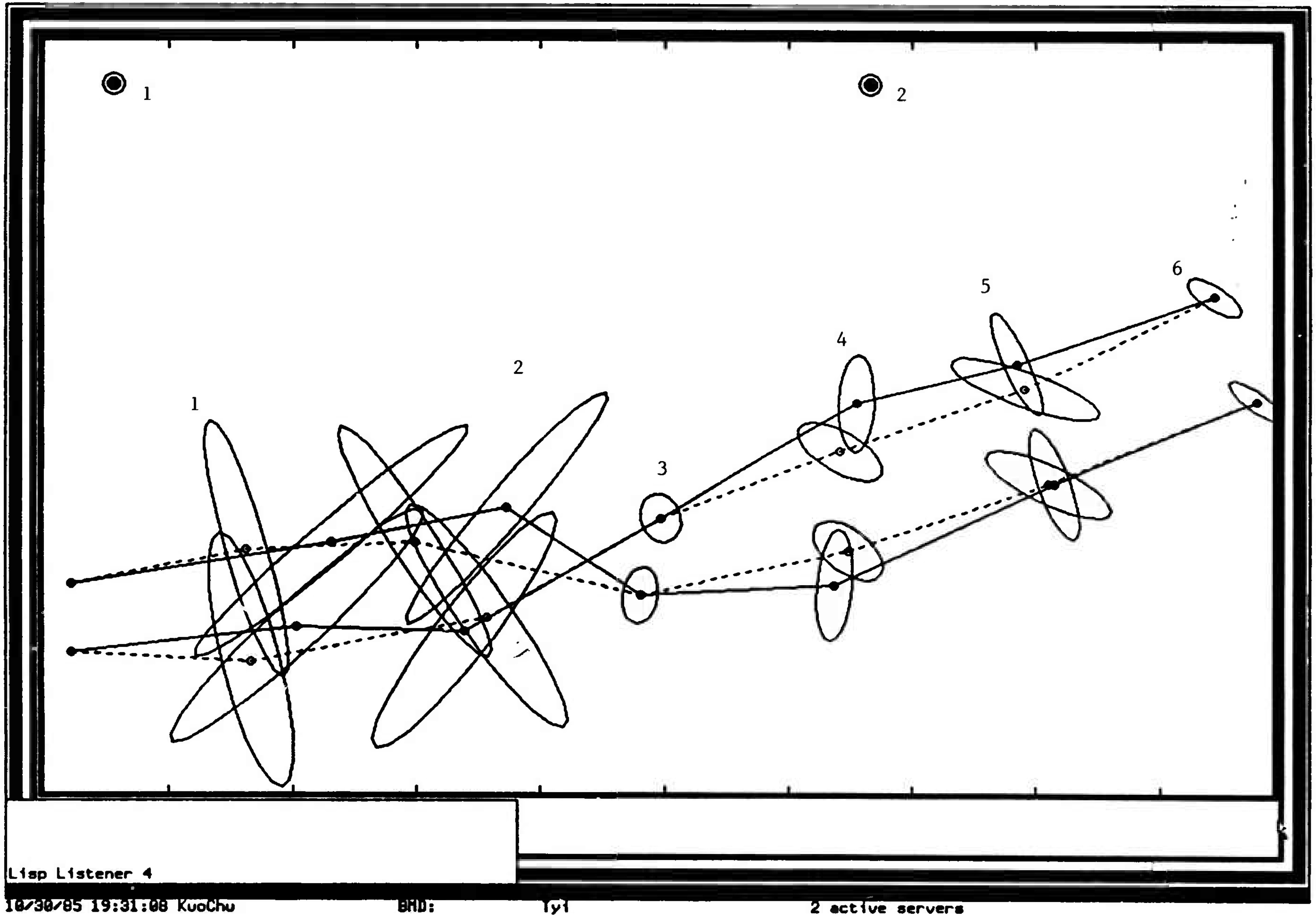


Figure 6-16: Example of Correct Hypothesis

accurate than the range. Initially for scan 1 and scan 2, each sensor has only one hypothesis consisting of two tracks. The tracks of Sensor 1 are denoted by solid lines with ellipses being the error covariances while the tracks of Sensor 2 are denoted by dotted lines. At time 3, the sensors broadcast to each other and information fusion takes place at each node. Because of the overlapping error ellipses, multiple hypotheses are formed. Figure 6-15 shows one hypothesis with two tracks and Figure 6-16 shows the other hypothesis. The pairs of tracks which are associated by the hypothesis can be traced from the centers of the fused tracks at time 3. From the two fusion hypotheses at time 3, each node processes additional measurements at scans 4 and 5. The two branches of the hypothesis tree are shown in the two figures. At time 6, the two nodes communicate again. Since in Figure 6-15, the ellipses of the same target according to the two nodes have little overlap, the probability of this hypothesis goes to zero. On the other hand, the hypothesis in Figure 6-16 is still valid and the track state estimates from the two nodes are fused to obtain an improved estimate.

6.4 CONCLUSION

We have found the simulation environment to be very useful in developing and evaluating algorithms and in studying the various issues associated with a DSN. The information fusion algorithms developed in Section 2 have been tested via simulations using various examples. The results demonstrate that the performance of the DSN nodes can be improved through communication. We have also shown that the multiple hypothesis approach developed in this research works better than the traditional (single hypothesis) approach when the scenario is complicated. Various communication schemes with different number of nodes have been examined. The simulation results have shown that the algorithms produce the expected performance.

7. CONCLUSIONS

The goals of our research were to further understand the issues associated with a distributed sensor network and to develop a general theory of distributed multitarget tracking to provide some guidance in building a DSN. This theory should be general enough to encompass arbitrary network structures, target and sensors models. For the theory to be relevant, it should also lead to implementable algorithms.

In our previous effort, we developed a general theory for tracking multiple targets. In the current effort, this theory was extended to the distributed situation. Information fusion algorithms were developed for fusing or integrating the information from other nodes with the local information. For general communication patterns, these information fusion algorithms insure that only consistent hypotheses are formed and that no information is double counted (which would lead to inconsistent conclusions). To the best of our knowledge the algorithms we have developed are the first to address these issues. The algorithms make heavy use of the so called *information graph* which can be viewed as an abstract model of the DSN communication structure. They also become the more standard algorithms with the appropriate assumptions.

In many military applications, targets frequently have special structures. For example, the state of a target may have different attributes such as location and velocity, a consistent set of features, etc. The sensors at the various DSN nodes may not be the same. One node may observe a certain set of features while another node may observe a different set of features so that the nodes need to cooperate to obtain a more global view and perhaps classify the target. By using the general results for distributed tracking, we have obtained algorithms for dissimilar sensors and targets with structured states. This is a case where our general theory applies quite readily.

Targets may also move in groups. This is another case where there is some structure on the targets. The knowledge about the group can be used in tracking since the individual targets no longer move independently. The key research issue is how to exploit this knowledge and avoid the combinatorics associated with tracking the members of the group. We have developed a mathematical framework for treating groups of targets. The results obtained to date are mostly for the centralized situation. Development of the distributed algorithms can follow the same approach used for independent targets.

To further evaluate the generality of our approach and also relate to the work undertaken by Lincoln Lab. on acoustic tracking, we applied our algorithms to acoustic sensors tracking low-flying targets. Because of the special characteristics of acoustic sensors (azimuth only measurements, propagation delays, etc.) some modifications had to be made to the track-level algorithms and likelihood computations. However, the general overall framework is still applicable. We also looked into the design of experiments using scenarios similar to those used by Lincoln Lab.

We have developed an environment for developing and demonstrating the algorithms. Scenarios for which we developed algorithms for include tracking land vehicles over a road network using MTI (moving target indicator) sensors and tracking air targets with sensors which are more accurate in azimuth measurements than range. Although not included in this report, the same algorithms were also used to track submarines.

In summary, we believe we have developed a theory which is applicable to general DSN problems. The theory has a sound theoretic basis and has also been demonstrated through simulation studies for different scenarios.

8. REFERENCES

[1] C.Y. Chong, S. Mori, E. Tse and R.P. Wishner, "Distributed Hypothesis Formation in Distributed Sensor Network," Final Report, A.I.& D.S. Technical Report TR-1015-02, Advanced Information & Decision Systems, Mountain View, CA, May 1983.

[2] S. Mori, C.Y. Chong, E. Tse and R.P. Wishner, "Multitarget Multisensor Tracking Problems - Part I: A General Solution and a Unified View on Bayesian Approaches," A.I.& D.S. Technical Report TR-1048-01, Advanced Information & Decision Systems, Mountain View, CA, June 1983 (revised Aug. 1984).

[3] S. Mori, C.Y. Chong, E. Tse and R.P. Wishner, "Tracking and Classifying Multiple Targets without *A Priori* Identification," *IEEE Trans. on Automat. Contr.,* Vol. AC-31, No. 5, pp 401-409, May 1986.

[4] C.Y. Chong, S. Mori, M.R. Fehling, K.C. Chang and R.P. Wishner, "Distributed Hypothesis Testing in Distributed Sensor Networks," A.I.& D.S. Interim Report TR-1048-02, Advanced Information & Decision Systems, Mountain View, CA, July 1984.

[5] J. Pearl, "Reverend Bayes on Inference Engines: A Distributed Hierarchical Approach," *Proc. of the Second Annual Conference on Artificial Intelligence*, Pittsburgh, PA, May 1982.

[6] D.B. Reid, "A Multiple Hypothesis Filter for Tracking Multiple Targets in a Cluttered Environment," Lockheed Report LMSC-D560254, Lockheed Palo Alto Research Laboratory, Palo Alto, CA, September 1977.

[7] D.B. Reid, "The Application of Multiple Target Tracking Theory to Ocean Surveillance," *Proc. of 18th IEEE Conf. Decision and Contr.r, Fort Lauderdale, FL, pp. 1046 - 1052, December 1979.*

[8] H.L. Wiener, W.W. Willman, I.R. Goodman and J.H. Kullback, "Naval Ocean Surveillance Correlation Handbook, 1978," NRL Report 8340, Naval Research Lab., Washington D.C., 1979.

[9] E. Flad, "Track Correlation for Aircraft Flying in Group Formation," *Siemens Forsch. -u. Entwickl. -Ber.*, Bd. 2, Nr. 6, 1973.

[10] S. Hale *et al.*, "Algorithm for Automatic Correlation," ATLAS Working Paper #13, P.A.& R.C., April 1978.

[11] R.J. Drazovich, S. Brooks and S. Foster, "Knowledge Based Ship Classification," *Proc. of Workshop of Image Understanding and Spacial Processing to Radar Signals for Automatic Ship Classification*, New Orleans, LA, February 1979

[12] D.S. Spain, G.L. Orr, C.A. O'Reilly and R.J. Drazovich, "Automated Multi-Sensor Unit Identification," Final Report, A.I.& D.S. Technical Report TR-3030-02, Advanced Information & Decision Systems, June 1984.

[13] "Synthetic Data Generator," M.I.T., Lincoln Laboratory, Lexington, MA, June 1983.

[14] R.R. Tenney and J.R. Delaney, "A Distributed Aeroacoustic Tracking Algorithm," *Proc. of the 1984 American Control Conference*, San Diego, CA, June 1984.

[15] R.T. Lacoss, "Distributed Sensor Network - Semiannual Technical Summary," ESD-TR-84-002, M.I.T Lincoln Laboratory, Lexington, MA, June 1984.

[16] K.C. Chang and Y. Bar-Shalom, "Joint Probabilistic Data Association for Multitarget Tracking with Possibly Unresolved Measurements and Maneuvers," *IEEE Trans. Automat. Contr.*, Vol. AC-23, pp. 585-594, July 1984.

APPENDIX A. MERGED MEASUREMENT LIKELIHOOD CALCULATION

The likelihood of a measurement y originating from two existing tracks, $\overline{\tau}_1$ and $\overline{\tau}_2$, is the joint mixture of probability density of y with the probability of event M of track merging and that of event D of target detection, and is expanded as

$$P(y, M, D \mid \overline{\tau}_1, \overline{\tau}_2) = \int P(y \mid M, D, x_1, x_2, \overline{\tau}_1, \overline{\tau}_2)\, P(M \mid D, x_1, x_2, \overline{\tau}_1, \overline{\tau}_2) \qquad \text{(A1)}$$

$$P(D \mid x_1, x_2, \overline{\tau}_1, \overline{\tau}_2)\, P(x_1 \mid \tau_1)\, P(x_2 \mid \tau_2)\, dx_1\, dx_2$$

When we identify D with the event in which $s_M^i \triangleq s_1 + w_s^i \geq s_{TH}$ for $i = 1$ and 2, we have

$$P(D \mid x_1, x_2, \overline{\tau}_1, \overline{\tau}_2) = P(D \mid s_1, s_2) = \left[1 - \operatorname{erf}\left(\frac{s_{TH} - s_1}{\sigma_s}\right)\right]$$

$$\left[1 - \operatorname{erf}\left(\frac{s_{TH} - s_2}{\sigma_s}\right)\right] \qquad \text{(A2)}$$

The track merging event is written as $M = \{ \mid \phi_M^1 - \phi_M^2 \mid < \delta\phi \}$, and hence we have

$$P(M \mid D, x_1, x_2, \overline{\tau}_1, \overline{\tau}_2) = \operatorname{erf}\left(\frac{\delta\phi - (\phi_1 - \phi_2)}{\sqrt{(\sigma_\phi^1)^2 + (\sigma_\phi^2)^2}}\right) - \operatorname{erf}\left(\frac{-\delta\phi - (\phi_1 - \phi_2)}{\sqrt{(\sigma_\phi^1)^2 + (\sigma_\phi^2)^2}}\right) \qquad \text{(A3)}$$

where σ_ϕ^i the standard deviation determined by Equation (5.4) for each i. The first factor in the integrand in (A1) is then written as

$$P(y \mid M, D, x_1, x_2, \overline{\tau}_1, \overline{\tau}_2) = P(\phi_M^m \mid M, \phi_1, \phi_2)\, P(s_M^m \mid D, s_1, s_2) \qquad \text{(A4)}$$

where ϕ_M^m and s_M^m are defined by Equations (5.5) - (5.7). In the first factor of the right hand side of (A4), the condition D was dropped because ϕ_M^m can be defined as being independent from the detection event D. Similarly, in the second factor, the condition M has been dropped because s_M^m can be considered to be defined by (5.7) regardless of whether or not the actual merging occurs. When we approximate (5.7) by (5.7'), we have

$$P(s_M^m \mid D, s_1, s_2, \bar{q}) = \frac{g\left(s_M^m - h_s^m(s_1, s_2; \bar{s}_1, \bar{s}_2) \; ; \; \sigma_s^m(\bar{s}_1, \bar{s}_2)\right)}{1 - \mathrm{erf}\left(\frac{s_{TH} - h_s^m(s_1, s_2; \bar{s}_1, \bar{s}_2)}{\sigma_s^m(\bar{s}_1, \bar{s}_2)}\right)} \tag{A5}$$

The denominator of the right hand side of (A5) is necessary because the range for s_M^m is $[s_{TH}, \infty)$. Furthermore, we may approximately equate the right hand side of (A2) with the denominator of the right hand side of (A5). Then, since the GTSD component and the SPTSD component of a track are independent from each other, we have

$$P(y, M, D \mid \bar{\tau}_1, \bar{\tau}_2) = \int P(\phi_M^m \mid M, \phi_1, \phi_2) \tag{A6}$$

$$\left[\mathrm{erf}\left(\frac{\delta\phi - (\phi_1 - \phi_2)}{\sqrt{(\sigma_\phi^1)^2 + (\sigma_\phi^2)^2}}\right) - \mathrm{erf}\left(\frac{-\delta\phi - (\phi_1 - \phi_2)}{\sqrt{(\sigma_\phi^1)^2 + (\sigma_\phi^2)^2}}\right)\right]$$

$$g\left(s_M^m - h_s^m(s_1, s_2; \bar{s}_1, \bar{s}_2) \; ; \; \sigma_s^m(\bar{s}_1, \bar{s}_2)\right) \; P(x_1 \mid \tau_1) \, P(x_2 \mid \tau_2) dx_1 dx_2$$

$$= \int P(\phi_M^m \mid M, \phi_1, \phi_2)$$

$$\left[\mathrm{erf}\left(\frac{\delta\phi - (\phi_1 - \phi_2)}{\sqrt{(\sigma_\phi^1)^2 + (\sigma_\phi^2)^2}}\right) - \mathrm{erf}\left(\frac{-\delta\phi - (\phi_1 - \phi_2)}{\sqrt{(\sigma_\phi^1)^2 + (\sigma_\phi^2)^2}}\right)\right] P(\phi_1 \mid \bar{\tau}_1) \, P(\phi_2 \mid \bar{\tau}_2) d\phi_1 d\phi_2$$

$$\int g\left(s_M^m - h_s^m(s_1, s_2; \bar{s}_1, \bar{s}_2) \; ; \; \sigma_s^m(\bar{s}_1, \bar{s}_2)\right) \; P(s_1 \mid \bar{\tau}_1) \, P(s_2 \mid \bar{\tau}_2) ds_1 ds_2$$

The last integral in (A6) can be easily calculated and yield to (5.21). On the other hand, since ϕ_M^m and the track merging event M are correlated, the calculation of the first integral in the last expression of (A6) is not so straightforward. But, according to [16], we have

$$\int P(\phi_M^m \mid M, \phi_1, \phi_2) \left[\mathrm{erf}\left(\frac{\delta\phi - (\phi_1 - \phi_2)}{\sqrt{(\sigma_\phi^1)^2 + (\sigma_\phi^2)^2}}\right) - \mathrm{erf}\left(\frac{-\delta\phi - (\phi_1 - \phi_2)}{\sqrt{(\sigma_\phi^1)^2 + (\sigma_\phi^2)^2}}\right)\right] \tag{A7}$$

$$P(\phi_1 \mid \bar{\tau}_1) \, P(\phi_2 \mid \bar{\tau}_2) d\phi_1 d\phi_2$$

$$= g\left(\phi_M - h_\phi^m(\bar{\phi}_1, \bar{\phi}_2 \; ; \; \bar{q}) \; ; \; \tilde{\sigma}_\phi^m\right) \left[\mathrm{erf}\left(\frac{\delta\phi - \tilde{\Delta}\phi}{\tilde{\sigma}_{\Delta\phi}}\right) - \mathrm{erf}\left(\frac{-\delta\phi - \tilde{\Delta}\phi}{\tilde{\sigma}_{\Delta\phi}}\right)\right]$$

which yields (5.17).